Escola de Ciências e Tecnologia · Departamento de Informática

Extracção de Informação de Documentos em Língua Portuguesa

José Luis Pinto Pedras

Orientador: Professor Doutor Paulo Quaresma (Universidade de Évora)

Évora, Outubro de 2010

UNIVERSIDADE DE ÉVORA

Escola de Ciências e Tecnologia · Departamento de Informática

Mestrado em Engenharia Informática

Extracção de Informação de Documentos em Língua Portuguesa

José Luis Pinto Pedras

Dissertação apresentada à Universidade de Évora para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Informática, realizada sob a orientação científica do Professor Doutor Paulo Quaresma (Universidade de Évora)

Évora, Outubro de 2010

Resumo

Com o aumento exponencial de informação disponível na Internet e devido ao facto da maioria desta estar num formato não estruturado, surgiu o conceito de Extracção de Informação cujo principal objectivo consiste na transformação da informação desorganizada e não estruturada num formato adequado aos sistemas informáticos.

Este trabalho incide sobre a Extracção de Informação de Documentos, mais precisamente na língua Portuguesa, sobre os quais é desenvolvido um sistema de extracção baseado em regras e padrões, e realizados testes comparativos entre o sistema e os principais métodos de aprendizagem automática (*Hidden Markov Model*, *Hidden Semi-Markov Model*, *Maximum Entropy Markov Model*, *Conditional Random Fields* e *Support Vector Machines*). O domínio utilizado é na área dos anúncios de venda de automóveis, cujos resultados obtidos são em média superiores a 90% para o sistema desenvolvido. Numa segunda fase são efectuados vários testes com conjuntos de documentos de diferentes dimensões no domínio dos anúncios de venda de casas, utilizando métodos de aprendizagem automática. Os resultados obtidos visam apurar as variações produzidas nas medidas de avaliação.

Abstract

Information Extraction from Portuguese Documents

With a exponential growth of available information on Internet, and due to most of that being in a non-structured format, has emerged the Information Extraction concept, which principal objective consists on transformation of unorganized and non-structured data to use in information systems.

This work is related with Information Extraction from Portuguese documents, where is developed a rules and patterns based extraction system, whose results are compared with machine learning methods (*Hidden Markov Model*, *Hidden Semi-Markov Model*, *Maximum Entropy Markov Model*, *Conditional Random Fields* e *Support Vector Machines*). The developed system achieved more than 90% (f-measure) in car sales listings domain. On a second stage tests, are used document sets with different dimensions using machine learning algorithms, in house sales listings domain, to evaluate changes on performance measures.

Agradecimentos

Iniciando pela secção académica, agradeço ao meu orientador, Professor Doutor Paulo Quaresma pelo apoio, acompanhamento e disponibilidade durante a realização deste trabalho.

Agradeço aos meus pais pelo apoio incondicional que me deram desde sempre e principalmente durante a minha carreira académica. Obrigado Pai e Mãe.

Conteúdo

Lista de Tabelas

Lista de Abreviaturas

Abreviatura	Descrição
CD	Conjunto de Documentos
CMM	Conditional Markov Model
CRF	Conditional Random Fields
EI	Extracção de Informação
HMM	Hidden Markov Model
HSMM	Hidden Semi-Markov Model
HTML	HyperText Markup Language
LALR	Look-Ahead Left to Right
MEMM	Maximum Entropy Markov Model
SGBD	Sistema de Gestão de Bases de Dados
SVM	Support Vector Machines
SVMCMM	Support Vector Machines Conditional Markov Model
VPCMM	Voted Perceptron Conditional Markov Model
VPHMM	Voted Perceptron Hidden Markov Model
VPSMM	Voted Perceptron Hidden Semi-Markov Model
XML	eXtensible Markup Language

Capítulo 1

Introdução

Com a explosão da popularidade e crescimento a larga escala da Internet também a quantidade de Informação disponível aumentou exponencialmente. Vinte e cinco mil milhões de páginas Web são o valor estimado de endereços indexados pelos principais motores de pesquisa no ano de 2009 (WorldWideWebSize, 2009). No entanto, apesar desta quantidade gigantesca de fontes de Informação disponíveis na Web, as mesmas sofrem de heterogeneidade e de falta de estrutura. Deste modo, o único método de acesso a tamanha quantidade de dados resume-se à navegação e pesquisa. A Extracção de Informação é um tipo de pesquisa documental que tem como principal objectivo a extracção de informação não estruturada a partir de fontes documentais e a posterior reestruturação desta em entidades, relações entre entidades e atributos descritores de entidades. Este método permite uma melhor pesquisa do que a procura por palavras-chave num conjunto não organizado. A estruturação da informação é um processo complexo que pode ser muito difícil de concretizar e é um desafio para muitas equipas de investigadores de todo o mundo que há duas décadas se dedicam à Extracção de Informação de Documentos.

A definição de Extracção de Informação é descrita pela autora Moens (2006), de uma forma simples e concisa: *"A Extracção de Informação é a identificação, classificação e estruturação em classes semânticas, de informação especifica encontrada*

em fontes não estruturadas, tais como documentos. O objectivo é a transformação num formato mais adequado a tarefas de processamento por parte dos sistemas de informação".

A Extracção de Informação divide-se em várias fases de acordo com a informação que se pretende extrair: a primeira fase denomina-se reconhecimento de nomes e entidades e define-se pelo reconhecimento e classificação de nomes em documentos (pessoas, localizações, companhias, entre outros); a segunda fase trata do reconhecimento de relações entre entidades, nomeadamente entre duas ou mais entidades; a terceira fase efectua o reconhecimento de regras semânticas entre os vários constituintes sintácticos da frase; a quarta e quinta fases tratam da resolução de co-referências entre nomes e pronomes e entre vários documentos do conjunto; por fim a sexta fase realiza a o reconhecimento de números e expressões temporais, mais precisamente a detecção de expressões temporais (relativas, absolutas e relativas a eventos) e a detecção e reconhecimento de números e a sua correspondente atribuição/resolução.

De modo a realizar as fases de extracção definidas anteriormente, um sistema de Extracção de Informação divide-se em dois processos de modo a tratar cada uma dessas fases. Deste modo, os processos gerais de um sistema de Extracção de Informação têm a seguinte apresentação: o processo inicial denomina-se *Tokenization* e consiste na partição do texto em unidades básicas (palavras, frases, parágrafos) que serão analisadas pelo sistema; o último processo analisa os dados obtidos anteriormente no âmbito do domínio definido aplicando o método de extracção e realizando a publicação dos resultados obtidos.

Os dois principais métodos utilizados na Extracção de Informação de Documentos denominam-se: métodos baseados em padrões e regras e métodos de aprendizagem automática. Inicialmente surgiram os métodos baseados em padrões e regras, que tal como o nome indica, utilizam uma base de conhecimento ou um conjunto de regras rígidas e padrões na Extracção de Informação de Documentos. Com o aumento da capacidade de processamento dos sistemas informáticos, desenvolveram-se

vários algoritmos de aprendizagem automática que na última década foram a principal área de estudo de várias equipas de investigadores.

1.1 Objectivos

O principal objectivo desta dissertação foi a Extracção de Informação de Documentos em língua Portuguesa e dividiu-se nos seguintes processos:

Desenvolvimento de um sistema de Extracção de Informação

Desenvolvimento de um sistema de extracção de entidades para um domínio definido utilizando as técnicas presentes no modelo de regras e padrões.

Comparação de resultados obtidos entre vários sistemas

Realização de um conjunto de testes comparativos entre o sistema desenvolvido e outros sistemas baseados em algoritmos de aprendizagem automática.

Estudo da variação da taxa de extracção entre conjuntos de documentos

Estudo e comparação de resultados para vários conjuntos de documentos de diferentes dimensões de um domínio definido, utilizando vários algoritmos de aprendizagem automática.

1.2 Metodologia

Os sistemas e testes desenvolvidos nesta dissertação dividiram-se nas seguintes fases:

- Escolha e preparação dos conjuntos de documentos a utilizar: escolha dos domínios e entidades; realização das anotações necessárias de modo a ser possível a obtenção de resultados.
- Desenvolvimento do sistema proposto:
  - Criação de uma base de conhecimento sobre o domínio.

- Definição dos padrões e regras a utilizar para cada entidade definida.
- Definição do sistema de resolução de conflitos entre regras.
- Comparação entre os resultados do sistema e os resultados anotados e obtenção das respectivas medidas de avaliação.

- Realização de testes com sistema de aprendizagem automática:
  - Utilização dos principais algoritmos de aprendizagem automática: *Hidden Markov Model*, *Hidden Semi-Markov Model*, *Maximum Entropy Markov Model*, *Conditional Random Fields* e *Support Vector Machines*.
  - Comparação dos resultados obtidos: com os do sistema desenvolvido para o mesmo domínio e conjunto de documentos; para conjuntos de documentos de diferentes dimensões dentro do mesmo domínio.

1.3 Estrutura do Documento

Esta dissertação encontra-se estruturada em 5 capítulos de acordo com o seguinte formato: no Capítulo 2 é apresentado o Estado da Arte cujo conteúdo é relacionado com os principais métodos, algoritmos e sistemas de Extracção de Informação; o Capítulo 3 apresenta a Arquitectura Proposta para a criação e desenvolvimento de um sistema de Extracção de Informação; o Capítulo 4 trata de um Estudo de Caso no qual se aplica o sistema desenvolvido na temática dos anúncios de venda de automóveis e casas; por fim, no Capítulo 5 apresentam-se as conclusões obtidas e propostas de ideias para o trabalho futuro.

Capítulo 2

Estado da Arte

A Extracção de Informação de Documentos tem como principal objectivo a extracção de informação não estruturada a partir de fontes documentais e a posterior reestruturação desta para utilização numa base de conhecimento. Inicialmente foram usados métodos baseados em regras que mais tarde, devido à sua fragilidade, evoluíram para métodos de aprendizagem automática.

Neste capítulo são apresentadas várias abordagens no que diz respeito aos formatos e sistemas baseados em padrões e regras, aos quais seguidamente se apresentam os principais métodos de aprendizagem automática. Por fim, são descritas as medidas de avaliação utilizadas na Extracção de Informação de documentos.

2.1 Métodos Baseados em Padrões e Regras

Os métodos de Extracção de Informação assentes em regras evoluíram ao longo do tempo. Os primeiros métodos de extracção eram baseados em dicionários (padrões) aos quais se seguiram a utilização de um conjunto de regras rígidas codificadas manualmente. Como a inserção de regras era uma tarefa muito demorada e tediosa desenvolveram-se algoritmos de aprendizagem automática que se ocupavam de criar as regras a partir de exemplos (Aitken, 2002).

Um sistema de extracção baseado em regras é essencialmente constituído por duas partes que compõem o método. A primeira parte é o conjunto de regras que vão ser aplicadas. A segunda parte do método é o sistema de controlo das regras existentes e definidas na primeira parte. Este sistema de controlo pode ser constituído de várias formas, tais como: o uso desordenado das regras e sobre as quais existe um sistema que resolve os conflitos quando existem incertezas; o uso de prioridades (ordenação) nas regras. O modelo de extracção baseado em regras tem como principal vantagem a possibilidade de consolidar ou aumentar o conjunto de regras de modo a obter melhores resultados.

Os sistemas baseados em regras consistem de um conjunto de regras gerais definidas que efectuam essa tarefa. Dentro dos sistemas de padrões e regras existem ainda dois conjuntos principais: os que utilizam regras definidas manualmente e os que contém métodos para aprender e adicionar novas regras ao sistema. As duas principais abordagens existentes nos sistemas automatizados de regras são: o método *bottom-up* que aplica as regras a partir de casos especiais para casos gerais e o método *top-down* que realiza essa tarefa em sentido contrário, ou seja, das regras gerais para as especiais.

A autora Sarawagi (2008), descreve a forma básica de uma regra no formato *padrão contextual* → *acção* em que o *padrão contextual* consiste no uso de um ou mais padrões etiquetados que capturam várias propriedades e contextos de uma ou várias entidades na forma como aparecem no documento. Os padrões podem ser uma expressão regular definida sobre as propriedades e contexto que os *tokens* apresentam no texto. A parte da *acção* é utilizada para aplicar os vários formatos gerais de regras definidos.

Um conjunto de regras para o reconhecimento simples de uma entidade consiste em três tipos de padrões:

- Um padrão (opcional) que captura a informação de contexto antes do inicio da entidade.

- O reconhecimento dos *tokens* da entidade através de padrões de reconhecimento.
- Um padrão (opcional) que captura a informação de contexto depois do fim da entidade.

Existem ainda outras abordagens para o caso de entidades mais complexas, em que se utilizam várias regras nas extremidades das entidades e todo o *interior* é considerado como uma entidade. São ainda utilizadas várias regras para extrair múltiplas entidades, em que uma ou mais regras fazem o reconhecimento das várias entidades simultaneamente.

A organização das regras implementadas é também uma importante tarefa num sistema de regras na medida em que grande parte do sucesso na Extracção de Informação depende da ordenação destas regras e da resolução de conflitos.

Existem várias estratégias para a ordenação e resolução de conflitos, das quais se destacam as duas mais relevantes:

- Tratamento das regras de uma forma desordenada e independente. Quando existem conflitos entre regras diferentes que cobrem o mesmo texto deve-se utilizar a regra que cobre maior parte do texto e em caso de empate escolher a com maior prioridade ou realizar a junção da informação quando as regras em conflito são para a atribuição da mesma entidade.
- Utilização de um sistema completo de prioridades em que todas as regras definidas apresentam uma prioridade. A escolha em caso de conflito recai sobre a de maior prioridade.

A maioria dos sistemas apresenta a necessidade de um conjunto muito elevado regras altamente precisas para realizar a tarefa de Extracção de Informação. A tarefa de construção de regras necessita de codificação por parte de um conjunto de pessoas muito experientes no domínio a que se aplica o sistema, o que pode ser

uma tarefa muito morosa. É com base nisto que se desenvolveram sistemas que efectuam a aprendizagem automática de regras a partir de documentos etiquetados para o efeito. O algoritmo de funcionamento da maioria dos sistemas que utilizam este método apresenta a seguinte definição (Sarawagi, 2008):

1. R = conjunto de regras, inicialmente vazio.
2. Enquanto existir uma entidade $x \in D$ que não tenha cobertura por uma qualquer regra de R.
   a) Criar novas regras sobre x.
   b) Adicionar as novas regras a R.
3. Realizar uma tarefa de refinamento de modo a retirar regras redundantes.

Para o método de criação de regras *bottom-up* o algoritmo tem a seguinte implementação:

1. Para cada tipo de etiqueta T
   a) Criação de uma regra semente para uma instância não abrangida. Uma regra semente consiste no uso de uma instância x que não tem qualquer regra associada na qual se colocam um conjunto de *tokens* w à esquerda e direita de $T \in x$ dando origem a uma regra no formato $x_{i-w}, ..., x_{i-1}x_i, ..., x_{i+w} \rightarrow T$ em que T corresponde à posição i de x.
   b) Generalização da regra semente
   c) Remoção das instâncias cobertas pelas novas regras

O modelo *top-down* têm uma abordagem diferente do anterior, iniciando-se a partir das regras especializadas. Deste modo, a sua definição é a seguinte: Seja R_0 a regra semente mais especializada a partir da qual se define a aprendizagem até $2w$ posições. É definido ainda um parâmetro s que indica a abrangência mínima de cada regra. O algoritmo apresenta a seguinte formulação:

1. R_1 = conjunto de regras de nível 1 que tem uma condição numa das $2w$ posições e uma abrangência superior a s.

2. Para um nível $L = 2$ até $2w$:

   a) R_L = Regras formadas pela intersecção de duas regras em R_{L-1} que são compatíveis em todas as condições de $L - 2$ excepto numa.

   b) Remoção das regras de R_L com abrangência menor que s

O resultado é um conjunto de regras que cobrem R_0 e têm uma abrangência de pelo menos s.

Depois de descrito o funcionamento dos sistemas baseados em padrões e regras, são apresentados alguns dos sistemas na área da Extracção de Informação que fazem uso destes métodos na sua implementação: ferramenta *FASTUS* que efectua a EI a partir de um sistema de regras predefinidas (Hobbs et al., 1993); sistema de criação de regras a partir de textos anotados denominada *CRYSTAL* (Soderland et al., 1995); criação e desenvolvimento da ferramenta *AutoSlog* e da sua evolução *AutoSlog-TS* que consiste num sistema de criação de regras a partir de documentos não etiquetados (Riloff, 1996); sistema de criação e aprendizagem de regras a partir de documentos semi-estruturados e não estruturados denominado *WHISK* (Soderland, 1999); ferramenta de criação automática de regras a partir de documentos estruturados e não estruturados denominada *BWI* (Freitag and Kushmerick, 2000); sistema de EI de páginas HTML que utiliza regras definidas numa linguagem declarativa denominada *Elog* e publica o resultado obtido no formato XML (Baumgartner et al., 2001); ferramenta de construção automática de regras baseada em padrões e com utilização de páginas na Internet (HTML) como fonte ao invés de conjuntos de documentos treinados (hui Chang and Lui, 2001); sistema de EI de entidades e relações em artigos biomédicos com recurso a padrões e analisadores sintácticos (Yakushiji et al., 2001); desenvolvimento e utilização de vários métodos, entre os quais um baseado em regras, na EI de entidades de mensagens de *voicemail* (Huang et al., 2001); utilização da linguagem declarativa *Prolog* para construção de regras

de EI de entidades (Emms, 2001); desenvolvimento de um sistema de criação de regras a partir de exemplos denominado $(LP)^2$ (Ciravegna, 2001); ferramentas de EI que utilizam um conjunto de regras declarativas cuja sintaxe é semelhante ao *SQL* (Jayram et al., 2006; Krishnamurthy et al., 2008).

2.2 Métodos de Aprendizagem Automática

Os métodos baseados em aprendizagem automática surgiram como forma de ultrapassar as debilidades do uso dos sistemas de regras no que diz respeito ao aumento do conjunto de regras que pode ser uma tarefa extremamente complicada e ao facto dos sistemas baseados em regras serem muito dependentes do domínio no qual se inserem. Ao longo dos anos, devido ao aumento do poder computacional, foram desenvolvidos vários métodos de aprendizagem automática.

Nesta secção, são apresentados vários algoritmos de aprendizagem automática, com inicio no *Hidden Markov Model*, ao qual seguidamente se apresentam duas evoluções distintas deste modelo, o *Hidden Semi-Markov Model* e o *Maximum Entropy Markov Model*, também denominado de *Conditional Markov Model*. O quarto algoritmo a ser analisado é o *Conditional Random Fields* e por fim o *Support Vector Machines*.

2.2.1 Hidden Markov Model (HMM)

A teoria dos *Hidden Markov Models (HMM)* na Extracção de Informação é um conceito com vários estudos realizados por várias equipas de investigadores da área. O modelo clássico do HMM foi publicado numa série de artigos no final da década de 60 e inicio de 70 pelos autores Baum and Eagon (1967); Baum et al. (1970); Baum and Petrie (1966). Ao contrário dos *Markov Models* em que são conhecidos os estados e os únicos parâmetros são as probabilidades de transição de estados, no HMM os estados não são directamente visíveis e apenas são conhecidos os resultados

de saída e não a sequência de estados que levou a essa saída. O funcionamento do HMM é descrito por Rabiner and Juang (2003) através da resolução de três problemas chave do modelo:

1. Dada uma sequência de observações $O = O_1, O_2, ..., O_T$, como calcular a probabilidade da sequência observada $P(O|\lambda)$?
2. Dada uma sequência de observações $O = O_1, O_2, ..., O_T$, como escolher a sequência de estados $I = i_1, i_2, ..., i_T$ óptima?
3. Como ajustar os parâmetros do modelo, de modo a maximizar $P(O|\lambda)$?

As respostas a estas questões e que descrevem o funcionamento do algoritmo HMM são as seguintes:

1. A resposta ao problema é dada pelo cálculo da probabilidade da sequência observada O dado o modelo λ através da enumeração de todas as sequências de estados de comprimento T. Para cada sequência fixa de estados $I = i_1 i_2 ... i_T$ a probabilidade da sequência observada O é $P(O|I, \lambda)$, onde:

$$P(O|I, \lambda) = b_{i1}(O_1) b_{i2}(O_2) ... b_{iT}(O_t)$$

Por outro lado, a probabilidade de uma sequência de estados I é dada por:

$$P(I|\lambda) = \pi_{i1} a_{i_1 i_2} a_{i_2 i_3} ... a_{i_{T-1} i_T}$$

Deste modo a probabilidade conjunta de O e I é dada por:

$$P(O, I|\lambda) = P(O|I, \lambda) P(I|\lambda)$$

Finalmente a probabilidade de O é obtida pela soma da probabilidade conjunta para as sequências de estados, ou seja:

$$P(O|\lambda) = \sum_{allI} P(O|I, \lambda) P(I|\lambda)$$

No entanto o cálculo desta probabilidade é uma tarefa com um custo temporal da ordem de $(2T-1)N^T$ (Rabiner and Juang, 2003) o que torna necessária uma nova abordagem no cálculo da probabilidade de O.

O procedimento implementado para resolver o custo temporal denomina-se método *Forward-Backward* (Rabiner, 1990; Rabiner and Juang, 2003) cujo autor refere que com este método o custo temporal obtido é da ordem N^2T, ou seja, muito menor do que utilizando o cálculo directo da probabilidade de O.

O algoritmo *Forward-Backward*, descrito por Hoberman and Durand (2006) no seu conjunto de notas têm a seguinte definição:

- O algoritmo Forward é um método dinâmico para calcular o valor de $P(O|\lambda)$. De notar que não se conhece a sequência de estados, ou seja, é necessário considerar todas as sequências existentes.

  Assim, o valor é obtido através do cálculo da probabilidade de estar no estado i após a geração da sequência até ao valor observado O_t. Este valor tem a seguinte definição:

  $$\alpha_t(i) = P(O_1, O_2, O_3, ..., O_t, q_t = S_i)$$

  O algoritmo tem a seguinte implementação:

  - Inicialização:

    $$\alpha_1(i) = \pi_i e_i(O_1)$$

  - Iteração:

    $$\alpha_{t+1}(i) = \sum_{j=1}^{N} \alpha_t(j) * a_{ji} * e_i(O_{t+1})$$

  - A probabilidade de observar toda a sequência é dada pela soma de todos os possíveis estados finais

    $$P(O) = \sum_{i=1}^{N} \alpha T(i)$$

- O algoritmo *Backward* funciona em sentido contrário do algoritmo *Forward*, ou seja, permite o cálculo do valor da probabilidade de O partindo do final da sequência. Seja $\beta_t(i)$ a probabilidade das últimas observações $T-t$ que terminam no estado i:

$$\beta_t(i) = P(O_{t+1}, O_{t+2}, O_{t+3}, ..., O_T | q_t = S_i)$$

Assim, o algoritmo tem a seguinte implementação:

- Inicialização:

$$\beta_T(i) = 1$$

- Iteração:

$$\beta_{t-1}(i) = \sum_{j=1}^{N} \beta_t(j) * a_{ij} * e_j(O_t)$$

- Começando em T e percorrendo até 1, calculando $\beta_t(i)$ obtém-se a probabilidade de observar toda a sequência

$$P(O) = \sum_{j=1}^{N} \pi_j e_j(O_1) \beta_1(j)$$

2. A resposta do problema 2 consiste na escolha da melhor sequência de estados. O método utilizado para encontrar essa sequência denomina-se algoritmo de *Viterbi* (Viterbi, 2003; Forney, 1973) cujos autores descrevem como um método recursivo para encontrar a solução ideal no problema da estimação da sequência de estados observado num processo de Markov.

Os autores Rabiner and Juang (2003) descrevem o funcionamento formal do algoritmo de *Viterbi* através da realização dos seguintes passos:

a) Inicialização:

$$\delta_1(i) = \pi_i b_i(O_1), \qquad 1 \leq i \leq N$$

$$\Psi_1(i) = 0$$

b) Passo recursivo:

$$\text{Para } 2 \leq t \leq T_t, \qquad 1 \leq j \leq N$$

$$\delta_t(j) = \max_{1 \leq i \leq N}[\delta_{t-1}(i)a_{ij}]b_j(O_t)$$

$$\Psi_t(j) = \text{argmax}_{1 \leq i \leq N}[\delta_{t-1}(i)a_{ij}]$$

c) Terminação:

$$P^* = \max_{1 \leq i \leq N}[\delta_T(i)]$$

$$i_T^* = \text{argmax}_{1 \leq i \leq N}[\delta_T(i)]$$

d) *Backtracking* da sequência de estados:

$$\text{Para } t = T-1, T-2, ..., 1$$

$$i_T^* = \Psi_{t+1}(i_{t+1}^*)$$

3. A resposta ao último problema consiste no ajuste dos parâmetros do modelo de modo a maximizar a probabilidade da sequência observada. O método que permite realizar esta tarefa é um método iterativo da família dos algoritmos *Expectation-Maximization (EM)* (Dempster et al., 1977) e denomina-se algoritmo *Baum-Welch* (Baum et al., 1970). Os autores Hoberman and Durand (2006) descrevem nas suas notas o funcionamento detalhado deste algoritmo.

   Dados os estados $O_1, O_2, ...,$ é necessário determinar os parâmetros que maximizam $\lambda = a_{ij}, e_i(.), \pi_i$, mas devido ao facto de ser intratável encontrar um máximo global é necessário enumerar todos os conjuntos de parâmetros λ_k e então calcular:

$$\text{Score}(\lambda_k) = \sum_d P(O^d|, \lambda_k) = \sum_d \sum_Q P(O^d|\lambda_k, Q)$$

   para cada λ_k.

Para uma dada sequência O^d a probabilidade de transitar de um estado i para j num dado tempo t é:

$$P(q_t^d = i, q_{t+1}^d = j|O^d, \lambda) = \frac{P(q_t^d = i, q_{t+1}^d = j, O^d)}{P(O^d)} = \frac{\alpha_t(i)a_{ij}e_j(O_{t+1}^d)\beta_{t+1}(j)}{P(O^d)}$$

A descrição dos termos da equação anterior são os seguintes:

- O termo $\alpha_t(i)$ representa a probabilidade do modelo ter emitido o conjunto de símbolos $O_1^d...O_t^d$ estando no estado S_i no tempo t. Este valor pode ser obtido utilizando o algoritmo *Forward*.
- Através do algoritmo *Backward* é possível obter o valor de $\beta_{t+1}(j)$ que é a probabilidade de emissão do resto da sequência estando no estado j no tempo $t+1$.
- Os termos a_{ij} e $e_j(O_{t+1}^d)$ indicam a probabilidade da transição de i para j e a emissão do valor em $t+1$

Assim é possível estimar:

$$A_{ij} = \sum_d \frac{1}{P(O^d)} \sum_t \alpha(t,i)a_{ij}e_i(O_{t+1}^d)\beta(t+1,i)$$

e

$$E_i(\sigma) = \sum_d \frac{1}{P(O^d)} \sum_{\{t|O_t^d=\sigma\}} \alpha(t,i)\beta(t,i)$$

Finalmente, a partir de A_{ij} e $E_i(\sigma)$ é possível re-estimar os parâmetros.

O algoritmo *Baum-Welch*, cujo funcionamento é semelhante aos algoritmos EM, ou seja, são atribuídos valores iniciais e estimada a verosimilhança dos dados cujo resultado é utilizado para re-estimar os parâmetros até ser obtido um máximo local. Deste modo, a sua definição é a seguinte:

a) Escolha de valores iniciais para $\lambda(\pi, a_{ij}, e_i(.))$

b) Determinação dos possíveis caminhos $Q^d = q_1^d, q_2^d, ...$

c) Contagem do possível número de transições A_{ij} do estado i até j, dada a estimativa de λ

d) Contagem de $E_i(\sigma)$, ou seja, a estimativa do número de vezes que foi emitido σ a partir do estado i

e) Estimação de $\lambda(\pi, a_{ij}, e_i(.))$ a partir de A_{ij} e $E_i(\sigma)$

f) Se não convergir, voltar ao passo 2

Quanto à Extracção de Informação, a utilização do modelo HMM foi efectuada em vários estudos dos quais se destacam: o sistema *Nymble* (Bikel et al., 1997) que utiliza o modelo HMM na EI de nomes e outras entidades não recursivas em textos, apresentando resultados de medida F superiores a 90% e que os autores referem como *"near-human performance"*; o sistema de EI de nomes de genes e localizações a partir de textos científicos (Leek, 1997); as evoluções ao modelo HMM (Freitag and Mccallum, 1999) que utilizam o conceito de *shrinkage* para aferir as probabilidades de emissão em conjuntos de treino de dimensão limitada diminuindo até 40% de taxa de erro relativamente aos sistemas anteriores; o sistema de EI de cabeçalhos de artigos científicos (Seymore et al., 1999) que utiliza o conceito de múltiplos estados por entidade, obtendo melhores resultados do que o uso de um estado apenas; o sistema de EI de restaurantes (nome, telefone e horário de funcionamento) (Zhang, 2001) a partir de ficheiros HTML; a ferramenta de EI de estilos bibliográficos (Connan and Omlin, 2000) que utiliza o HMM para extrair o estilo das referências no documento e a partir das quais efectua a extracção da informação de entidades (autores, titulo, ano de publicação, etc...); o programa de EI de anúncios de emprego (Au and Cheung, 2004); o sistema de reconhecimento de entidades na língua Coreana (Yun, 2010) cuja definição morfológica e sintáctica das palavras numa entidade está relacionada com as palavras circundantes e não pertencentes a essa entidade.

2.2.2 Hidden Semi-Markov Model (HSMM)

O *Hidden Semi-Markov Model (HSMM)* é visto como uma extensão ao modelo definido anteriormente, o *Hidden Markov Model*. A principal diferença entre os dois

modelos reside no facto do HSMM definir-se pelo uso de uma cadeia semi-escondida de Markov assim como pelo facto de cada estado ter uma duração variável. Outra diferença importante é o facto de no HMM ser assumida uma observação por estado enquanto no HSMM cada estado pode emitir uma ou várias observações sequenciais.

O modelo geral do *Hidden Semi-Markov Model (HSMM)* descrito por Yu (2009) evidência o funcionamento deste método. Assim, além da notação definida para o algoritmo HMM assume-se também a duração d de um estado dado. O número de observações efectuadas enquanto no estado i é determinada pelo tempo gasto nesse estado (d). Assumindo uma cadeia escondida de Markov discreta temporalmente com um conjunto de estados $S = 1, ..., M$. A sequência de estados define-se por $S_{1:T} \triangleq S_1, ..., S_T$ onde $S_t \in S$ é o estado no momento t. Denota-se a sequência de observações $O_{1:T} \triangleq O_1, ..., O_T$ onde $O_t \in \nu$ é observável num momento temporal t e $\nu = v_1, v_2, ..., v_k$ o conjunto de valores observáveis. Para a sequência de observações $O_{1:T}$, a sequência de estados subjacente é dada por $S_{1:d_1]} = i_1, S_{[d_1+1:d_1+d_2]} = i_2, ..., S_{[d_1+...+d_{n-1}+1:d_1+...+d_n]} = i_n$ e as transições de estado são $(i_m, d_m) \rightarrow (i_{m+1}, d_{m+1})$ para $m = 1, ..., n-1$, onde $\sum_{m=1}^{n} d_m = T, i_1, ..., i_n \in S$, e $d_1, ..., d_n \in D$. Define-se a probabilidade de transição de estados de $(i, d') \rightarrow (j, d)$ para $i \neq j$ por:

$$a_{(i,d')(j,d)} \triangleq P[S_{[t+1:t+d]} = j | S_{[t-d'+1:t]} = i]$$

sujeita a $\sum_{j \in S \setminus \{i\}} \sum_{d \in D} a_{(i,d')(j,d)} = 1$ com probabilidade zero na transição para o mesmo estado, $a_{(i,d')(i,d)} = 0$, onde $i, j \in S$ e $d, d' \in D$. A partir desta definição verifica-se que o estado anterior a i iniciou-se em $t - d' + 1$ e terminou em t com a duração d' e transitou para o estado j com duração d de acordo com a probabilidade de transição de estado $a_{(i,d')(j,d)}$. O estado j começará em $t + 1$ e terá término em $t + d$, ou seja, quer o estado quer a duração são dependentes do estado e duração anteriores. No estado j serão efectuadas d observações $o_{t+1:t+d}$ emitidas. Define-se a probabilidade da emissão por:

$$b_{j,d(O_{t+1:t+d})} \triangleq P[O_{t+1:t+d} | S_{t+1:t+d} = j]$$

que se assume independente no tempo t. Seja a distribuição inicial do estado

$$\pi_{j,d} \triangleq P[S_{[t-d+1:t]} = j], \qquad t \leq 0, d \in D$$

que representa a probabilidade do estado inicial e a sua duração antes de $t = 1$, ou antes da obtenção da primeira observação o_1. Então, o conjunto de parâmetros do modelo HSMM denota-se:

$$\lambda \triangleq a_{(i,d')(j,d)}, b_{j,d(v_{k_1:k_d})}, \pi_{i,d}$$

onde, $i, j, \in S$, $d, d' \in D$ e $v_{k1:kd}$ representa $v_{k1}...v_{kd} \in \nu \times ... \times \nu$.

Os métodos de cálculo dos vários valores definidos anteriormente são muito semelhantes aos definidos no HMM. Deste modo, o HSMM utiliza uma extensão dos algoritmos *Forward-Backward* (Rabiner, 1990; Rabiner and Juang, 2003) e *Viterbi* (Viterbi, 2003; Forney, 1973) bem como uma variação do modelo EM (Dempster et al., 1977) na estimação dos parâmetros.

Quanto à utilização deste algoritmo, o autor Yu (2009) evidencia algumas das áreas em que foi aplicado, nomeadamente, no reconhecimento de actividade humana, na detecção de anomalias em tráfego de redes, no reconhecimento e síntese de discursos, na segmentação de imagens, entre outros.

No que diz respeito à EI de documentos, este algoritmo foi aplicado principalmente na área de reconhecimento de texto impresso e manuscrito. Destas aplicações destacam-se as seguintes: reconhecimento de texto e expressões (Chen et al., 1993; Cai and Liu, 1999), quer na língua inglesa, quer noutras linguagens (Safabakhsh and Adibi, 2005; Benouareth et al., 2006); EI de artigos biomédicos em formato digital (George R. Thoma and Misra, 2005) para construção de uma base de conhecimento composta por um conjunto de meta dados sobre a informação extraída.

2.2.3 Maximum Entropy Markov Model (MEMM)

O *Maximum Entropy Markov Model (MEMM)*, também denominado de *Conditional Markov Model* é visto como uma extensão ao modelo *Hidden Markov Model*. A

principal diferença entre os dois modelos reside no facto de o MEMM utilizar o conceito de maximização da entropia (Jaynes, 1957).

Os autores Mccallum and Freitag (2000) apresentam o modelo do MEMM, a partir do modelo HMM. Assim, no MEMM as funções de transição e observação definidas no HMM são substituídas por uma função apenas $P(s|s', o)$ que se denomina a probabilidade do estado actual s, dado o estado anterior s' e a observação actual o. Também, em contraste com o HMM no qual as observações apenas dependem do estado actual, no MEMM podem depender do estado anterior.

Uma das alterações deste método relativamente ao HMM reside no facto de redefinir alguns métodos definidos anteriormente, como por exemplo, o algoritmo *Forward-Backward*. O *Forward*, definido no HMM como a probabilidade de obter uma sequência de observações até um determinado tempo t, redefine-se:

$$\alpha_{t+1}(s) = \sum_{s' \in S} \alpha_t(s') * P_{s'}(s|o_{t+1})$$

O algoritmo *Backward* é também redefinido e representa-se por:

$$\beta_t(s') = \sum_{s \in S} P(s|s', o_t) * \beta_{t+1}(s)$$

O conceito de máxima entropia é utilizado para estimar distribuições de probabilidade a partir de dados. É baseado no princípio que o melhor modelo para os dados é o que tem melhor consistência com certas restrições derivadas dos dados de treino. Cada restrição dispõe de alguma característica dos dados de treino que deve ser presente na distribuição. Essas restrições pode ser representadas por várias características binárias, como por exemplo, *"a observação começa por uma letra maiúscula"* ou *"a observação é um número"*, entre outros.

De notar que as características não dependem apenas da observação mas também do resultado previsto pela função a ser modelada. Deste modo, de cada característica a obtém-se uma função $f_a(o, s)$ em que o é a observação actual e s o possível novo estado actual. Assim, e definindo cada a como um par $a = \langle b, s \rangle$ em que b é uma

característica binária e s é o estado de destino, obtém-se:

$$f_{\langle b,s\rangle}(o_t, s_t) = \begin{cases} 1, & \text{se } b(o_t) \text{ é verdade, e } s = s_t \\ 0, & \text{caso contrário} \end{cases}$$

Formalmente, para cada estado anterior s' e característica a a função de transição $P'_s(s|o)$ deve obedecer à propriedade

$$\frac{1}{m_{s'}} \sum_{k=1}^{m_{s'}} f_a(o_{t_k}, s_{t_k}) = \frac{1}{m_{s'}} \sum_{k=1}^{m_{s'}} \sum_{s \in S} P_{s'}(s|o_{t_k}) f_a(o_{t_k}, s)$$

A distribuição de máxima entropia que satisfaz essas restrições é única (Pietra et al., 1997), e tem a seguinte fórmula exponencial:

$$P_{s'}(s|o) = \frac{1}{Z(o, s')} \exp\left(\sum_a \lambda_a f_a(o, s)\right)$$

em que λ_a são os parâmetros que se pretendem estimar e $Z(o, s')$ o factor de normalização.

O método iterativo utilizado para estimar os valores de λ_a que formam a solução denomina-se *Generalized Iterative Scaling (GIS)* (Darroch and Ratcliff, 1972). A definição do algoritmo GIS, para aprender a função de transição $P_{s'}$ para o estado s', é composta pelos seguintes passos:

1. Cálculo dos dados médios de treino de cada característica a

$$F_a = \frac{1}{m_s} \sum_{k=1}^{m_s} f_a(o_{t_k}, s_{t_k})$$

2. Inicio da iteração 0 do GIS com uns quaisquer valores arbitrários para os parâmetros

3. Na iteração j, usar o valor corrente de $\lambda_a^{(j)}$ em $P_{s'}^{(j)}(s|o)$ para calcular o valor esperado de cada característica

$$E_a^{(j)} = \frac{1}{m_s} \sum_{k=1}^{m_s} \sum_{s \in S} P_{s'}^{(j)}(s|o_{t_k}) f_a(o_{t_k}, s)$$

4. Mudar cada λ_a de modo a que o valor estimado de cada característica se aproxime da média dos dados de treino correspondentes

$$\lambda_a^{(j+1)} = \lambda_a^{(j)} + \frac{1}{C}\log(\frac{F_a}{E_a^{(j)}})$$

5. Até ser obtida a convergência pretendida, voltar ao passo 3

O funcionamento do MEMM é então descrito no seguinte conjunto de itens que resume a sua implementação:

- *Início:* Uma sequência de observações $o_1, ..., o_m$, a que corresponde uma sequência de etiquetas $l_1, ..., l_m$. Um conjunto de estados, cada um com uma etiqueta, uma estrutura de transição sujeita a restrições e um conjunto de características observáveis no estado.
- Determinar a sequência de estados associada à sequência de etiquetas observada.
- Depositar os pares observação - estado (s, o) nos estados anteriores s' correspondentes, como dados de treino para cada função de transição de estados $P_{s'}(s|o)$.
- Obter a solução com máxima entropia para a função discriminativa de cada estado através do algoritmo GIS.
- *Fim:* O modelo do MEMM que utiliza uma sequência de observações não etiquetadas e realiza a correspondente previsão dessas etiquetas.

Quanto à utilização deste algoritmo na EI, existem vários estudos desenvolvidos por vários autores dos quais se evidenciam os seguintes: EI de artigos da *Usenet*, nomeadamente do cabeçalho, corpo, e perguntas-respostas (Mccallum and Freitag, 2000); EI de nomes e organizações em várias linguagens (Holandês e Espanhol) com recurso ao algoritmo MEMM (Jansche, 2002); identificação e extracção de nomes e entidades em artigos biomédicos (feng Lin et al., 2004; Dingare et al., 2005; Kim

et al., 2005); utilização de um sistema híbrido que engloba o algoritmo MEMM e um conjunto de regras na EI de entidades (Fresko et al., 2005); realização de vários testes com utilização de vários conjuntos de documentos e métodos de etiquetagem do conjunto de treino na EI de entidades (Krishnan and Ganapathy, 2005); EI de documentos através do desenvolvimento de um novo algoritmo que utiliza as vantagens do MEMM e do HMM (Li et al., 2009); reconhecimento de nomes de pessoas em textos antigos de fontes em mau estado (Packer et al., 2010a) utilizando o algoritmo MEMM entre outros;

2.2.4 Conditional Random Fields (CRF)

Os *Conditional Random Fields* foram desenvolvidos como uma melhoria às debilidades existentes nos algoritmos *Hidden Markov Model* e *Maximum Entropy Markov Model* no que diz respeito ao problema de *label bias* (as transições a partir de um estado competem apenas contra as desse estado ao invés de todas as transições existentes no modelo) (Lafferty, 2001). O mesmo autor descreve o funcionamento do CRF, através do seguinte conjunto de definições:

Seja X uma variável aleatória sobre um conjunto de dados a ser etiquetado e Y a variável aleatória sobre as sequências de etiquetas correspondentes. Todos os componentes Y_i de Y pertencem a um domínio finito γ. Por exemplo, na EI de entidades X corresponde ao texto sobre o qual se efectua a extracção, Y corresponde às entidades presentes nesse texto e γ todas as entidades definidas no modelo.

Seja $G = (V, E)$ um grafo tal que $Y = (Y_v)_{v \in V}$, no qual Y é indexado pelos vértices de G. Então, (X, Y) é o *conditional random field* no caso em que quando condicionado em X, as variáveis aleatórias Y_v obedecem à propriedade de *Markov* no que diz respeito ao grafo, $p(Y_v|X, Y_w, w \neq v) = p(Y_v|X, Y_w, w \sim v)$, onde $w \sim v$ significa que w e v são vizinhos em G.

Comparativamente ao algoritmo MEMM definem-se um conjunto de características que expressam alguma informação nos dados de treino que deve ser utilizada no mo-

delo. As características podem ser do tipo *estado* $s(y_1, x, i)$ ou *transição* $t(y_{i-1}, y_b, x, i)$ onde y_{i-1} e y_i são os estados, x a sequência de entrada e i a posição de entrada. Quando i é 1 (estado inicial da sequência), $t(y_{i-1}, y_b, x, i) = 0$. Abaixo são apresentados dois exemplos de características:

$$s_j(y_1, x, i) = \begin{cases} 1 \text{ se a observação na posição } i \text{ é a palavra 'olá'} \\ 0 \text{ caso contrário} \end{cases}$$

$$t_j(y_{i-1}, y_b, x, i) = \begin{cases} 1 \text{ se } y_{t-1} \text{ tem a etiqueta 'nome' e } y_t \text{ a etiqueta 'título'} \\ 0 \text{ caso contrário} \end{cases}$$

As funções de características dependem então do estado actual (função de *estado*) ou do estado actual e anterior (função de *transição*). É possível utilizar uma função de características f_j global, que pode ser uma função de *estado*, $s_j(y_1, x, i) = s_j(y_{i-1}, y_1, x, i)$ ou uma função de *transição* $t_j(y_{i-1}, y_1, x, i)$, que se representa por $f_j(y_{i-1}, y_1, x, i)$.

O vector global $F_j(x, y)$ para a sequência de entrada x e sequência de estados y define-se:

$$F_j(x, y) = \sum_{i=1}^{T} f_j(y_{i-1}, y_1, x, i)$$

Considerando k funções de características a distribuição de probabilidade condicional definida pelo CRF é:

$$P(y|x) = \frac{1}{Z} \exp(\sum_{j=1}^{k} \sum_{i=1}^{T} \lambda_j f_j(y_{i-1}, y_i, x, i))$$

ou

$$P(y|x) = \frac{1}{Z_x} \exp(\sum_{j=1}^{k} \lambda_j F_j(x, y)$$

onde Z representa um factor de normalização cuja definição é:

$$Z = \sum_{y \in Y} \exp(\sum_{j=1}^{k} \lambda_j F_j(x, y))$$

Dada a equação anterior, a sequência com maior probabilidade para um conjunto de observações de entrada x define-se:

$$y_* = \arg\max_y P(y|x)$$

que pode ser eficientemente calculada utilizando uma variação do algoritmo de *Viterbi* (Viterbi, 2003; Forney, 1973). O cálculo da probabilidade das transições utiliza também uma variação do algoritmo *Forward-Backward* (Rabiner, 1990; Rabiner and Juang, 2003) definido anteriormente.

Os parâmetros podem ser obtidos através do uso de um estimador de máxima verosimilhança devido ao facto do CRF ter muitas das vantagens do MEMM, entre as quais, uma função de verosimilhança convexa que garante que o processo de treino converge para o máximo global (Peng and McCallum, 2006).

Deste modo, algoritmos como o *Generalized Iterative Scaling* ou *Improved Iterative Scaling* (Pietra et al., 1997) podem ser utilizados para o treino do CRF. No entanto, foi provado que o algoritmo *quasi-Newton gradient-climber* têm uma conversão mais rápida (Malouf, 2002; Sha and Pereira, 2003).

Devido ao facto do CRF apresentar vantagens em relação aos métodos baseados em cadeias de *Markov* foi utilizado com sucesso por vários investigadores na área da EI, dos quais se destacam os seguintes trabalhos: evolução do algoritmo CRF e comparação com o modelo inicial na EI de entidades (McCallum, 2003); sistema de EI de entidades a partir de tabelas (Pinto et al., 2003); EI de contactos e redes sociais a partir de documentos de correio electrónico e Internet (Culotta et al., 2004; Minkov and Wang, 2005); ferramenta de EI de entidades em artigos biomédicos (Settles, 2004; Mcdonald and Pereira, 2005; Ponomareva et al., 2006; Klinger et al., 2007); sistema de EI de entidades a partir de artigos científicos (Peng and McCallum, 2006); utilização do CRF na EI de documentos na língua chinesa (Wu and Zhou, 2008); ferramenta de EI de entidades e relações em redes sociais académicas (investigadores e publicações) (Tang et al., 2008); comparação entre algoritmos e sistemas na EI de nomes de pessoas a partir de fontes com *ruído* (Packer et al., 2010b); sistema de EI

de entidades num conjunto de documentos clínicos (Dalianis and Velupillai, 2010).

2.2.5 Support Vector Machines (SVM)

Os *Support Vector Machines* (Cortes and Vapnik, 1995), definido como um classificador é utilizado na EI na medida em que a sua aplicação permite analisar dados e reconhecer padrões. Como o SVM é um classificador, dado um conjunto de treino anotado com categorias, ele constrói um modelo e aplica-o a novos dados (não treinados) identificando a que categoria pertencem.

O algoritmo SVM, descrito por Moens (2006) divide-se em várias fases: numa primeira fase é realizada a definição para dados linearmente separáveis; na segunda fase é generalizado para dados não separáveis; na terceira e última fase é demonstrado o uso de funções de *kernel* para dados que não se podem representar por decisão linear.

Dado um conjunto de características, numa classificação linear, cujo objectivo é obter o melhor plano que divide essas características nas classes pretendidas. Assim, esse plano define-se: dado o conjunto S com n exemplos $S = (x_1, y_1), ..., (x_n, y_n)$, onde $x_1 \in \Re^p$ (espaço dimensional p) e $y_1 \in \{-1, 1\}$ que indica que x_1 é positivo ou negativo, respectivamente. Quando se treina um conjunto de dados linearmente separáveis assume-se que existe algum plano que os separa em positivos e negativos. Os pontos existentes nesse plano satisfazem $\langle w \cdot x_i \rangle + b = 0$, onde w representa a direcção perpendicular(normal) ao plano e b o valor que move o plano paralelamente a si próprio. A distância perpendicular entre o plano e a origem define-se por $\frac{|b|}{||w||}$, onde $||w||$ representa a norma Euclidiana de w.

Seja $d_+(d_-)$, a distância mais curta que separa o plano do valor positivo(negativo), define a margem do plano. O objectivo é então encontrar o plano com a maior margem possível.

Dado um conjunto de dados linearmente separáveis e que satisfazem as seguintes

restrições:

$$\langle w \cdot x_i \rangle + b \geq 1 \qquad \text{para } y_i = 1$$

$$\langle w \cdot x_i \rangle + b \leq -1 \qquad \text{para } y_i = -1$$

que se combinam na inequação

$$y_i(\langle w \cdot x_i \rangle + b) - 1 \geq 0 \qquad \text{para } i = 1, ..., n$$

Os planos que definem as margens representam-se:

$$H_1 : \langle w \cdot x_i \rangle + b = 1$$

$$H_2 : \langle w \cdot x_i \rangle + b = -1$$

cujas distâncias à origem são:

$$d_{H_1} = \frac{|1 - b|}{||w||}$$

$$d_{H_2} = \frac{|-1 - b|}{||w||}$$

onde, $d_+ = d_- = \frac{1}{||w||}$ e a margem $= \frac{2}{||w||}$

De modo a maximizar a margem, é aplicada a seguinte função:

Minimizar $w, b : \langle w \cdot w \rangle$

Sujeito a: $y_i(\langle w \cdot x_i \rangle + b) - 1 \geq 0, \qquad i = 1, ..., n$

Devido ao facto de ser complicado aplicar inequações de restrições, é introduzido o uso de *multiplicadores de Lagrange* λ_i. A função resultante é a seguinte:

$$\text{Maximizar: } W(\lambda) = \sum_{i=1}^{n} \lambda_i - \frac{1}{2} \sum_{i,j=1}^{n} \lambda_i \lambda_j y_i y_j \langle x_i \cdot x_j \rangle$$

$$\text{Sujeito a: } \lambda_i \geq 0, \qquad \sum_{i=1}^{n} \lambda_i y_i = 0, i = 1, ..., n$$

Como são apenas realizados *produtos internos* entre os exemplos de treino, pode-se expressar a função como uma combinação linear de pontos. Resolvendo o problema de optimização quadrática é obtida a seguinte função de decisão $h(x)$:

$$h(x) = \text{sign } (f(x))$$

$$f(x) = \sum_{i=1}^{n} \lambda_i y_i \langle x_i \cdot x \rangle + b$$

A função anterior apenas depende de vectores de suporte em que $\lambda_i > 0$, ou seja, apenas os exemplos de treino que são vectores de suporte têm influência na função de decisão.

Caso os exemplos sejam linearmente separáveis, mas a sua separação contenha alguns erros (*ruído*), é possível redefinir o modelo anterior de modo a que seja tido em conta a ocorrência de erros ξ_i, cuja soma não pode exceder um valor predefinido. Assim, os planos que definem as margens representam-se:

$$H_1 : \langle w \cdot x_i \rangle + b = 1 - \xi_i$$

$$H_2 : \langle w \cdot x_i \rangle + b = -1 + \xi_i$$

E a respectiva função que maximiza a margem:

$$\text{Minimizar } \xi, w, b : \langle w \cdot w \rangle + C \sum_{i=1}^{n} \xi_i$$

$$\text{Sujeito a: } y_i(\langle w \cdot x_i \rangle + b) - 1 + \xi_i \geq 0, \qquad i = 1, ..., n$$

em que $\sum_{i=1}^{n} \xi_i$ representa o erro por classificação errada e C o factor *peso*.

Quando se classifica dados de linguagem nem sempre é possível separar linearmente os dados. Nesse caso é necessário mapear esses dados para uma dimensão onde sejam linearmente separáveis. No entanto, trabalhar com grandes dimensões leva a problemas de ordem computacional. Deste modo, o uso de funções de *kernel* é necessário na medida em que estas funções projectam os dados para uma dimensão mais elevada na qual é mais fácil a separação linear. Formalmente uma função de *kernel* K define-se:

$$K(x_i, x_j) = \sum_{k} \phi_(x_i)\phi_k(x_j) = \langle \phi(x_i) \cdot \phi(x_j) \rangle$$

Por outras palavras uma função de *kernel* é um produto interno num outro espaço (potencialmente muito complexo). A função de *kernel* tem de preencher uma série de requisitos, ou seja, tem de ser simétrica $|K(x_i, x_j) = K(x_j, x_i)|$ e definida positivamente. Deste modo, a matriz $n \times n$ definida por $G_{ij} = K(x_i, x_j)$,

denominada de matriz de *kernel* também o é. Dada a matriz G é então encontrado o plano com maior margem que separa as instâncias das diferentes classes.

Assim, substitui-se o produto interno pelo *kernel* correspondente na função de decisão e fica:

$$h(x) = \text{sign}(f(x))$$

$$f(x) = \sum_{i=1}^{n} \lambda_i y_i \langle \phi(x_i) \cdot \phi(x) \rangle + b = \sum_{i=1}^{n} \lambda_i y_i K(x_i, x) + b$$

No que diz respeito ao uso do SVM na EI, existem várias soluções implementadas das quais se destacam as seguintes: EI de entidades utilizando o algoritmo SVM com várias variações de modo a obter a melhor relação extracção/tempo (Isozaki and Kazawa, 2002); ferramenta de EI de entidades em artigos biomédicos com utilização do algoritmo SVM (Kazama et al., 2002; Takeuchi and Collier, 2003; Yang et al., 2010; Saha et al., 2010); desenvolvimento de uma ferramenta para EI de entidades de um conjunto de artigos científicos (Giles et al., 2003); EI de entidades com utilização de um conjunto de documentos do domínio das informações de actos terroristas (Sun et al., 2003); apresentação de um sistema de EI de entidades com uso de vários conjuntos de documentos (apresentação de seminários e anúncios de emprego), realização de testes comparativos com outros algoritmos e ferramentas de EI (Li et al., 2005); sistema de EI de nomes, organizações e expressões temporais a partir de textos escritos na língua Grega (Lucarelli et al., 2007); realização de um conjunto de testes com utilização de várias abordagens (regras e aprendizagem automática), entre as quais o SVM na EI de entidades (Mansouri et al., 2008); desenvolvimento de um sistema que utiliza o algoritmo SVM na EI de entidades em tabelas de um conjunto de documentos sobre mutações genéticas (Wong et al., 2009);

2.3 Medidas de Avaliação

De modo a avaliar a performance de um sistema de EI é utilizado um conjunto de medidas de avaliação. O sistema apresentado por Makhoul et al. (1999) visa uniformizar o cálculo destas medidas de modo a tornar comparáveis os resultados obtidos pelos vários sistemas. Assim, definem-se as seguintes variáveis:

- C - Número de termos correctos, que estão presentes na hipótese e validados na referência.
- S - Número de termos incorrectos, que estão na hipótese e na referência mas são errados.
- D - Número de termos existentes na referência que não figuram na hipótese
- I - Número de termos existentes na hipótese que não figuram na referência

Deste modo, as medidas de avaliação têm a seguinte representação:

- **Precisão:** Trata dos erros de substituição e de inserção.

$$\text{Precisão} = \frac{C}{C+S+I}$$

- **Abrangência:** Trata dos erros de substituição e de eliminação.

$$\text{Abrangência} = \frac{C}{C+S+D}$$

- **Medida F:** Também denominada de *F-score, F-measure ou F1-score*, define-se pela média harmónica da precisão e da abrangência.

$$\text{Medida F} = \frac{2 * \text{Precisão} * \text{Abrangência}}{\text{Precisão} + \text{Abrangência}}$$

Capítulo 3

Arquitectura Proposta

Neste capítulo são apresentadas as ferramentas utilizadas na EI de documentos de acordo com as duas grandes categorias de sistemas: os sistemas baseados em conjuntos de regras, que utilizam um conjunto de regras para a extracção das entidades pretendidas; os sistemas baseados em aprendizagem automática, que utilizam um conjunto de treino a partir do qual é aplicado um algoritmo de EI e gerado um conjunto de anotações que seguidamente são aplicadas no conjunto de documentos efectuando a EI para as entidades treinadas.

Assim, no primeiro subcapítulo é apresentada a ferramenta ExtrAuto que utiliza um sistema de regras, as suas funcionalidades e interface e a descrição do funcionamento interno no que diz respeito às regras utilizadas na EI. No segundo subcapítulo é realizada a descrição do programa MinorThird (Cohen, 2004a) que é baseado num sistema de aprendizagem automática.

3.1 Programa ExtrAuto

Para a realização da extracção de informação de anúncios de automóveis desenvolvi uma aplicação baseada num sistema de regras de nome ExtrAuto.

Esta aplicação é implementada na linguagem Java e permite realizar a extracção de informação das entidades usuais presentes no domínio dos anúncios de venda de automóveis, mais precisamente, a marca, o modelo, o preço, a cilindrada, a potência, o ano do registo do veículo, a data do anúncio, a localização da venda, o número do anúncio, o número de quilómetros efectuados, a lotação do veículo, o número de portas, o nome do anunciante, os contactos do anunciante (telefone, telemóvel, email e endereços da Internet), a existência e validade da inspecção periódica obrigatória e o combustível do veículo.

Para além das entidades definidas anteriormente são também extraídos vários equipamentos presentes num veiculo nomeadamente, o abs, o esp, os vidros eléctricos, o ar condicionado, o rádio, as jantes de liga leve, o fecho centralizado, a direcção assistida, os espelhos eléctricos, o alarme, os bancos reguláveis em altura, o volante regulável em altura, o airbag, o cruise control, os sensores de estacionamento, o sistema de navegação, os bancos em pele, os faróis de nevoeiro, o imobilizador electrónico, os bancos aquecidos, os espelhos aquecidos, os vidros escurecidos, os bancos com regulação eléctrica, os faróis xénon, o tecto de abrir, os bancos desportivos, a pintura metalizada, os faróis com regulação em altura, os sensores de chuva, o computador de bordo, o spoiler traseiro, os bancos traseiros rebatíveis, as barras no tejadilho, o sensor de luminosidade, a suspensão desportiva, o relógio, o conta-rotações, o lava-faróis, o sistema isofix, o apoio de braço, a capota eléctrica nos veículos cabriolet, os encostos de cabeça traseiros, o volante com regulação em profundidade e o volante desportivo.

Depois de identificadas as entidades que o programa permite extrair é descrito na próxima subsecção como se processa o funcionamento interno do programa e quais as regras gerais que permitem efectuar a extracção de uma entidade.

3.1.1 Funcionalidades do programa

O programa ExtrAuto realiza na tarefa de EI duas funções distintas de acordo com as preferências do utilizador.

Seguidamente à inicialização do programa é efectuada a selecção das directorias que contém os ficheiros com os anúncios de venda de automóveis (CD), das anotações realizadas para o CD utilizado e da colocação de ficheiros gerados pelo programa.

A interface do programa é bastante simples conforme se pode verificar pela figura 3.1 e a partir da qual se realizam as escolhas sobre as várias funcionalidades disponíveis. Todas as mensagens de execução do programa são mostradas na janela o que permite uma fácil intervenção em caso de falhas ou resultados não previstos.

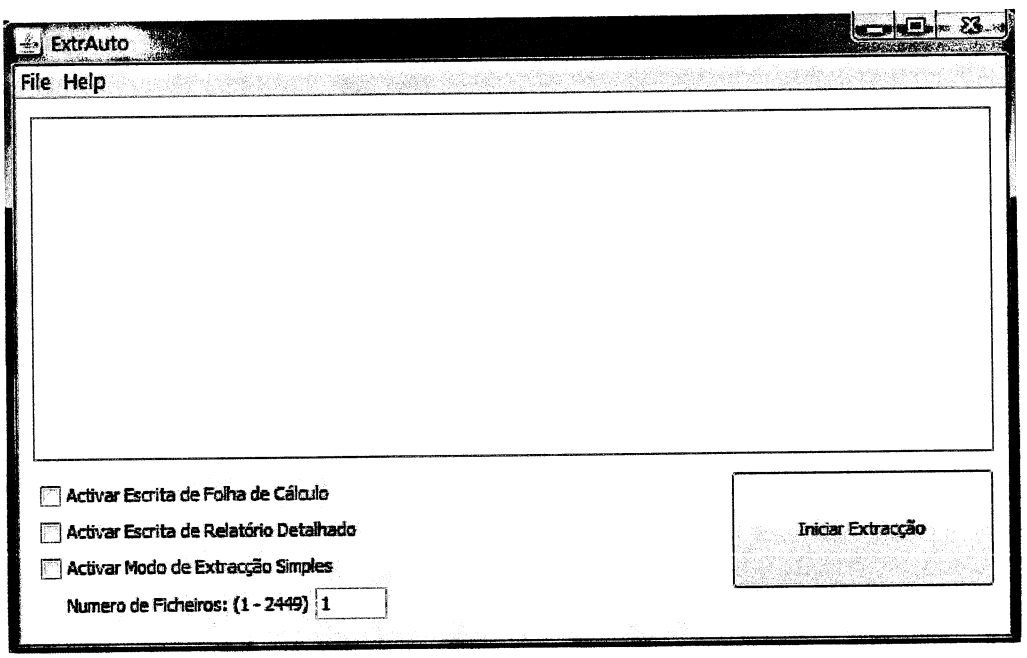

Figura 3.1: Interface do programa ExtrAuto

A primeira funcionalidade implementada é a de extracção única ou simples que consiste na EI da pasta definida com o CD e apresenta os resultados na janela de execução ou num ficheiro de texto caso a opção "Activar Escrita de Relatório Detalhado"esteja activada. Esta funcionalidade dispõe ainda de um sistema de escolha aleatória dos documentos presentes no CD que permite que o conjunto escolhido

pelo programa contenha um número de elementos definido na caixa "Número de Ficheiros"e sem que existam elementos repetidos. No caso de colocação o valor 0 o programa efectua a extracção de todos os ficheiros disponíveis na pasta.

A segunda funcionalidade consiste na realização da tarefa de EI à qual se segue a comparação com o conjunto de anotações e consequente escrita de resultados num ficheiro de folha de cálculo para posterior análise.

Esta tarefa é activa quando não está seleccionada a opção "Activar Modo de Extracção Simples"permitindo ainda a escolha das opções "Activar Escrita de Folha de Cálculo"(não disponível na extracção simples) e "Escrita de Relatório Detalhado"que escreve num ficheiro de texto os resultados da tarefa de EI.

Os ficheiros anotados são parte essencial desta segunda funcionalidade pois é a partir deles que é efectuada a tarefa de EI. Um ficheiro anotado é constituído por um formato próprio parecido com as etiquetas HTML que o divide em várias partes estruturadas, nomeadamente, no número do ficheiro no CD, no anúncio a que corresponde e nas anotações realizadas.

As anotações efectuadas num ficheiro de anotação são também elas próprias providas de um formato estruturado que permite a identificação da categoria a que pertencem (informações do veículo, combustível e equipamentos do veículo), da entidade que a identifica e do valor anotado para essa entidade que é variável na categoria das informações/combustível e que apresenta o valor afirmativo/negativo "SIM/NÃO"na categoria equipamentos.

Assim, a segunda funcionalidade implementada processa-se da seguinte forma:

- O programa verifica quais os ficheiros existentes na pasta das anotações e carrega para memória as entidades e valores anotados;
- Para cada um dos ficheiros anotados presentes na estrutura que suporta as informações recolhidas é efectuada a tarefa de EI para o anúncio correspondente (identificado pelo seu número único);

- A ultima fase deste processo é variável de acordo com as opções escolhidas pelo utilizador. A opção mais usual é a de activar a caixa "Activar Escrita de Folha de Cálculo"que efectua a escrita dos resultados obtidos num ficheiro do programa Microsoft Excel 2000/XP/2003 (*.xls). Neste ficheiro é atribuída a cada entidade uma folha com a restrição de as primeiras três folhas conterem as seguintes informações: a folha um é composta por informações sobre o CD; a folha dois contém os dados anotados dispostos numa tabela em que a cada entidade corresponde uma coluna e a cada linha o ficheiro anotado; a folha três é semelhante à folha dois mas contém os dados extraídos pelo programa. Nas restantes folhas (uma por entidade) os dados obedecem ao seguinte formato: a primeira coluna é composta pelos dados das anotações; a segunda coluna pelos dados extraídos.

A escolha de uma folha de cálculo para a colocação dos dados anotados/extraídos é efectuada por duas razões principais: para a realização de comparações de resultados e cálculo de medidas (abrangência, precisão e medida F); para ser facilitada a inserção dos dados extraídos na base de dados que suporta a tarefa de EI, pois actualmente existem várias ferramentas de suporte aos SGBD que permitem a inserção de dados a partir de folhas de cálculo.

Depois de apresentado o funcionamento para o utilizador do programa ExtrAuto é tratado na próxima subsecção o funcionamento interno do programa, mais precisamente do sistema de EI, e das regras de extracção das entidades que o constituem.

3.1.2 Funcionamento do sistema de EI

Antes da descrição do sistema de EI é necessário evidenciar o funcionamento da base de conhecimento que serve de suporte ao programa.

A BD é constituída por vários ficheiros com informação relacionada com o domínio em estudo (anúncios de venda de automóveis). Um dos ficheiros constituintes é a lista de vocábulos admitidos como nomes próprios na nação Portuguesa,

disponibilizada pelo Instituto dos Registos e do Notariado e presente no sítio de Internet da instituição em `http://www.irn.mj.pt`.

O ficheiro principal da BD é o que identifica os grupos e entidades e para cada um destes elementos contém as palavras que o descrevem no anúncio. Este ficheiro está definido de uma forma que permite a fácil adição de novas palavras ao sistema possibilitando o refinamento da EI. O formato definido neste ficheiro é definido pela expressão `[grupo] | [entidade] > [palavra 1] # ... # [palavra x] #` em que a cada entidade corresponde uma linha.

Por exemplo, para a entidade marca a linha do ficheiro que a contém representa-se pela seguinte expressão:

```
info | Marca > amc # aro # acura # alfa romeo # alpina # aston
martin # audi # austin # bmw # bentley # buick # ccxr # cadillac
# caparo # caterham # chevrolet # chrysler # citroën # citroen
# cobra # corvette # dacia # daewoo # daihatsu # datsun # dodge
# ferrari # fiat # ford # freightliner # gmc # graber # heinkel
# honda # hummer # hyundai # ihc # iveco # infiniti # isuzu #
jaguar # jeep # kia # lada # lamborghini # lancia # land rover
# lexus # ligier # lotus # mcc # mg # mack # maserati # matra
# mazda # mercedes # mercedes benz # mercedes  benz # mercury #
mini # mitsubishi # morgan # nsu # nash # nissan # oldsmobile #
opel # peugeot # pontiac # porsche # proton # renault # rolls
royce # royce # rover # saab # seat # skoda # smart # ssangyong
# subaru # suzuki # tvr # talbot # tata # toyota # triumph # umm
# vauxhall # volga # volkswagen # volvo # wartburg # yugo # vw
#
```

Conforme se pode verificar na definição da entidade estão presentes quase todos os construtores mundiais de automóveis, incluindo também marcas clássicas já

extintas (Matra, Talbot, Umm, Yugo, etc...) e marcas exclusivas/raras.

Terminada a definição da BD de suporte ao programa descreve-se abaixo o funcionamento do sistema na realização da tarefa de EI de anúncios de venda de automóveis.

A primeira fase da tarefa de EI do programa ExtrAuto é definida pelo uso de um analisador LALR para extracção de entidades. Foram utilizados as seguintes ferramentas para gerar o código Java necessário ao programa:

- Analisador lexical JFlex disponível em `www.jflex.de` e descrito como um gerador de código Java para utilização com um analisador LALR.
- Programa CUP disponível em `http://www2.cs.tum.edu/projects/cup/` e que permite a definição de um analisador LALR a partir de um conjunto de regras. Esta ferramenta utiliza o código gerado pelo programa JFlex para criar os ficheiros que implementam o LALR definido e gerar código final na linguagem Java que permite a sua integração com a aplicação desenvolvida.

O uso de um analisador sintáctico tem como principal função a extracção das entidades preço, localização, data do anúncio e número do anúncio. A máquina de estados cujas regras criam o analisador LALR utilizado estão representadas na figura 3.2. As entidades definidas anteriormente estão especificadas no CD de modo a ser possível a utilização desta ferramenta na tarefa de EI.

Para além das entidades obrigatórias é também possível extrair mais algumas entidades com esta ferramenta dependendo da sua presença e localização nos documentos do CD. Ou seja, o analisador LALR desenvolvido está apto a extrair a marca, o modelo, o ano do veículo, a cor, o número de quilómetros e a potência.

A segunda fase da tarefa de EI consiste na extracção das entidades pertencentes ao grupo dos contactos. As entidades que pertencem a este grupo são: os números de telefone e telemóvel; os endereços de correio electrónico; os endereços da Internet presentes no anúncio.

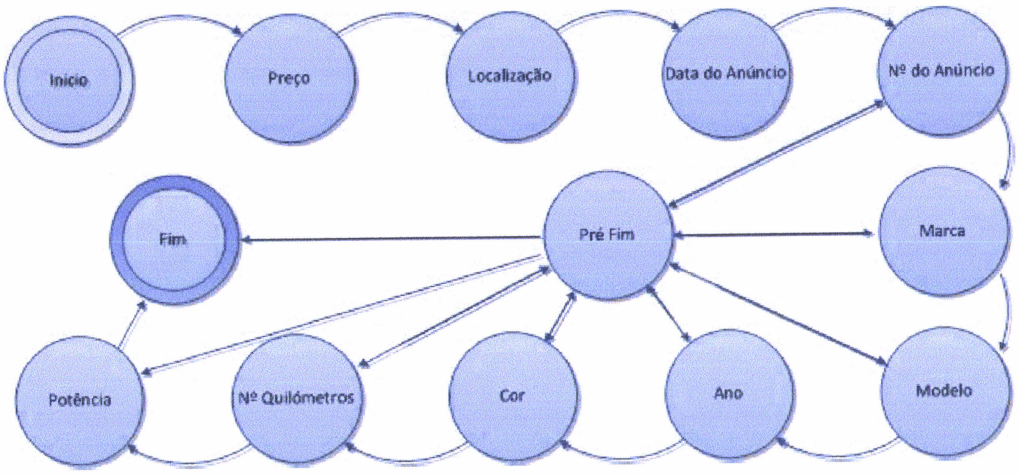

Figura 3.2: Máquina de estados finitos do parser LALR do programa ExtrAuto

O fluxograma presente na figura 3.3 representa como se processa esta fase. De notar que as entidades pertencentes ao grupo contactos são definidas apenas por uma palavra.

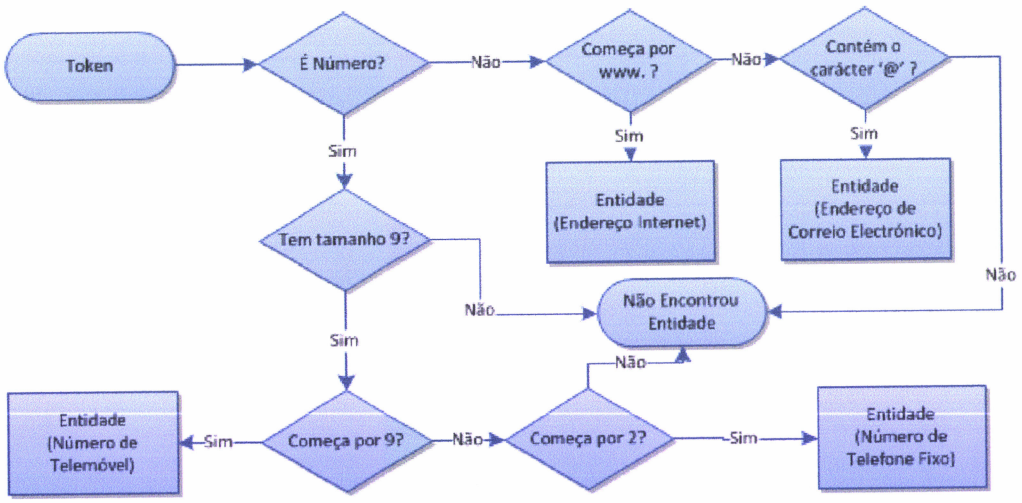

Figura 3.3: Fluxograma da EI do grupo contactos.

Conforme se pode verificar no fluxograma a extracção das entidades do grupo dos contactos é baseada num conjunto de regras para aferir a qual entidade pertence a palavra em análise. A primeira regra utilizada é a verificação se a palavra é um número ou não, o que permite separar entre as entidades números de telefone/telemóvel e as restantes. Após a utilização da regra anterior, para a palavra

que não é um número, devido à especificidade das entidades endereço de correio electrónico (utilização do caracter '@' ou da sub-palavra "[at]") e endereço da Internet (início da palavra pela expressão "www."ou "http://") é facilmente identificável uma palavra pertencente a estas duas entidades.

A palavra que na primeira regra é identificada como um número é submetida a uma regra de dimensão de modo a aferir se possui o número de caracteres correspondente a um número de telefone/telemóvel (em Portugal são utilizados nove dígitos) e no caso de sucesso desta regra a uma de verificação de qual o tipo de dispositivo (telefone e telemóvel) através dos caracteres iniciais da palavra, ou seja, um número de telemóvel em Portugal é identificado no início do seu número pelos valores "96", "91", "93"e "92"enquanto um número de telefone identifica-se pelo valor "2".

Na terceira e última fase são tratados os restantes grupos de entidades (equipamento e informações). Para melhor se compreender o sistema de EI para estas entidades que podem ter na sua constituição uma ou mais palavras é apresentado na figura 3.4 o fluxograma que evidência o mecanismo de procura para uma ou mais palavras.

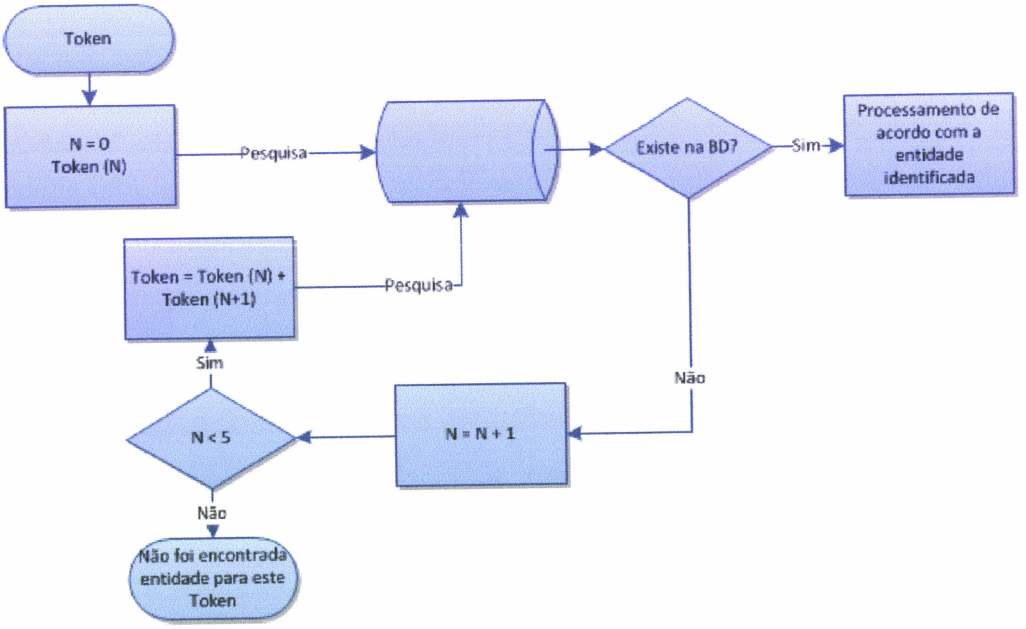

Figura 3.4: Fluxograma da identificação de palavras pertencentes a uma entidade.

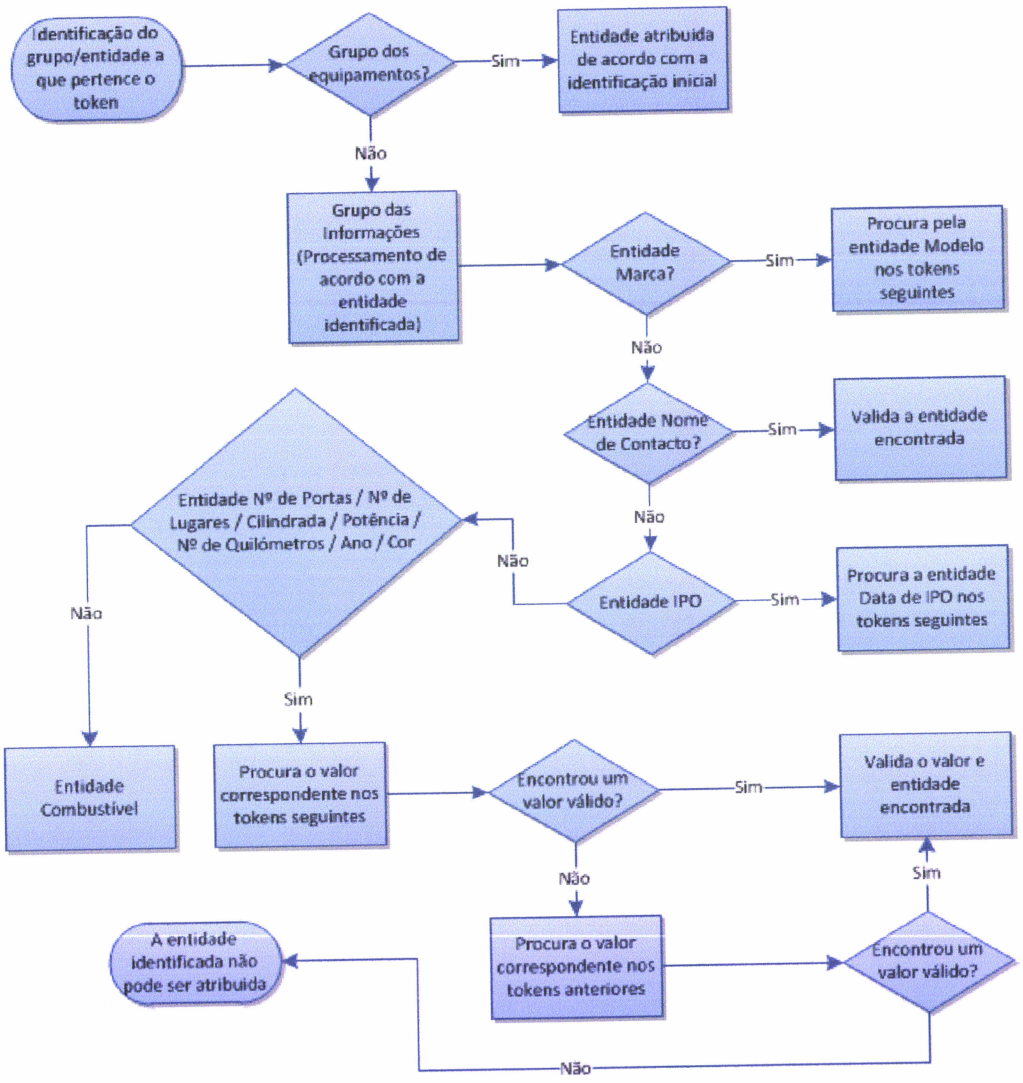

Figura 3.5: Fluxograma de EI para uma palavra identificada na BD

As entidades inspecção periódica obrigatória (IPO) e a sua data estão ligadas entre si e são as próximas a serem processadas. Quando é identificada na BD uma referência ao IPO é realizada a atribuição à entidade correspondente. A entidade data do IPO é pesquisada nas palavras seguintes até um máximo de cinco palavras. Durante esta pesquisa é verificada a existência de um conjunto mês/ano ou apenas ano através de validação das palavras. Quando não é encontrada qualquer valor para a entidade data do IPO é apenas atribuída a entidade IPO à palavra em processamento.

A próxima fase de processamento é igual para um conjunto de entidades, nomeadamente, o número de portas e lugares, a cilindrada, a potência, o número de quilómetros, o ano e a cor. A verificação destas entidades baseia-se no seguinte formato: quando é encontrada uma entidade pertencente a este grupo é efectuada uma verificação nas palavras seguintes do valor correspondente. Se esse valor for válido é atribuído à entidade correspondente. No caso de não se encontrar um valor nas palavras seguintes é efectuada uma procura nas palavras anteriores e efectuada a respectiva validação. Em caso de sucesso é realizada a atribuição e no caso contrário é processada a palavra seguinte pois apesar de ser sido encontrada uma referência a uma entidade esta não tem um valor válido.

A validação dos valores encontrados é efectuada de uma forma diferenciada de acordo com a entidade identificada. Os valores de validação estão contidos num ficheiro de configuração podendo ser alterados a qualquer momento. Assim para cada entidade são utilizados os seguintes valores:

- Na entidade número de portas os valores possíveis pertencem ao intervalo 2 a 5.
- Para a entidade número de lugares os valores estão no intervalo de 2 a 9.
- Na entidade cilindrada o valor tem de estar compreendido entre 500 e 8000, ou no caso de ser utilizado o formato convertido para decímetros cúbicos, entre 0,5 e 8,0.

- Para a entidade potência os valores de validação são entre 30 e 700.
- A entidade número de quilómetros apresenta valores entre 500 e 300000.
- O ano do veículo têm valores de validação no intervalo 1950 até ao ano corrente(2010).
- Os valores de validação da entidade cor estão presentes na BD e são verificados através de pesquisa.

A última fase de processamento pertence à atribuição da entidade combustível. O processo é semelhante ao das entidades do grupo dos equipamentos, ou seja, é feito de acordo com a identificação na BD.

Por fim são apresentadas as entidades atribuídas para cada documento de acordo com as escolhas do utilizador na interface gráfica.

Após a descrição da ferramenta ExtrAuto é apresentado no subcapítulo seguinte o programa MinorThird (Cohen, 2004a) que utiliza algoritmos de aprendizagem automática na EI.

3.2 Programa MinorThird

O programa MinorThird (Cohen, 2004a) define-se como uma ferramenta que permite realizar a categorização e extracção de documentos. Está disponível em `http://minorthird.sourceforge.net/` e apresenta vários modelos de extracção e classificação automática de documentos dos quais se destacam os seguintes:

- Classificação
  - K Nearest Neighbors (KNN)
  - Voted Perceptron (VP)
  - Support Vector Machines (SVM)

  - Naive Bayes
  - Decision Tree's
  - Maximum Entropy

- Extracção
  - HMM (Hidden Markov Model)
  - CMM (Conditional Markov Model)
  - CRF (Conditional Random Fields)
  - SMM (Hidden Semi-Markov Model)
  - MEMM (Maximum Entropy Markov Model)

Relativamente aos modelos de algoritmos de EI descritos acima a ferramenta MinorThird implementa um conjunto de algoritmos de modo a obter a melhor relação tempo/qualidade de extracção. Assim, são implementados os seguintes algoritmos: VPHMM (Collins, 2002) que utiliza o *Voted Perceptron* para estimar os parâmetros do HMM; VPCMM, semelhante ao anterior mas para o algoritmo CMM; VPSMM e VPSMM2 (Cohen, 2004b) que utilizam o *Voted Perceptron* para estimar os parâmetros do SMM; SVMCMM, que utiliza o modelo SVM para o algoritmo CMM; MEMM (Mccallum and Freitag, 2000); CRF (Lafferty, 2001; Sha and Pereira, 2003); SemiCRF (Sarawagi and Cohen, 2004).

De modo a utilizar as várias funcionalidades do programa de uma forma simplificada é disponibilizada uma interface gráfica que permite a escolha das tarefas a realizar (classificação ou extracção) existindo dentro de cada categoria uma interface de escolha de qual o tipo de acção (treino; teste e resultados; treino seguido de teste e resultados) a efectuar.

Depois de escolhida a acção a realizar é apresentada uma janela com várias opções e uma consola de resultados. Das várias opções destaca-se a de escolha/configuração do método a utilizar que permite preparar uma tarefa de EI, nomeadamente, na escolha do CD a treinar, na entidade a extrair, na escolha do CD a testar e no

algoritmo a utilizar (cada algoritmo dispõe ainda de uma série de configurações de acordo com a sua implementação).

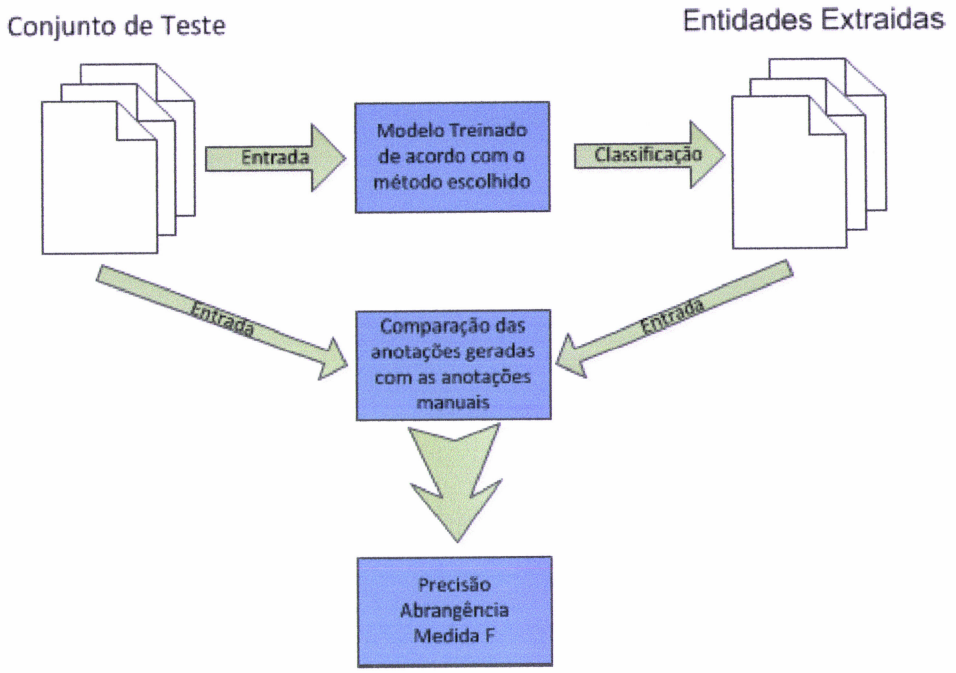

Figura 3.6: Funcionamento do programa MinorThird

O método de funcionamento do programa para uma tarefa completa de EI (treino e teste) descreve-se pela figura 3.6 que mostra como se processa a EI. Cada uma das fases é definida por:

- Fase de Treino: O CD previamente anotado é treinado para a entidade e algoritmo escolhidos e classificado de modo a obter um conjunto de dados anotados pelo programa.
- Fase de Teste: É realizada a comparação de resultados entre o conjunto anotado e o conjunto que se pretende testar. De notar que o conjunto de teste têm de estar anotado para que o programa efectue esta fase com sucesso. Por fim são apresentados os resultados dos testes ao utilizador, dos quais se destacam a abrangência, a precisão e a medida F.

Para identificar quais as entidades a extrair o programa utiliza um formato de marcação com etiquetas XML, ou seja, cada palavra(as) existentes no documento e que descrevem uma entidade são inseridas dentro da etiqueta que indica as palavras e a entidade a que pertencem. Por exemplo, um documento de entrada com a frase:

"Paulo da Gama acompanhou o seu irmão Vasco da Gama na descoberta do caminho marítimo para a Índia "

e no qual se pretende anotar a entidade *nome* apresenta o seguinte formato de entrada no programa:

```
<nome>Paulo da Gama</nome> acompanhou o seu irmão <nome>Vasco da
Gama</nome> na descoberta do caminho marítimo para a Índia
```

O modelo de extracção gerado a partir das entidades no programa MinorThird utiliza um método de classificação dos termos existentes num texto. Assim, começa por gerar um conjunto de características para cada um dos termos do documento. Por exemplo, na frase anterior e utilizando o método para extrair a entidade *nome* são marcados como POS todos os termos que pertencem à entidade e NEG aos que não pertencem. De acordo com o algoritmo escolhido podem existir ainda outras categorias. Por exemplo, os termos POS no algoritmo CRF dividem-se em três categorias: BEGIN que indica o termo inicial de uma entidade a extrair; CONTINUE que é associado a um termo que da entidade que não é o inicial nem o final; END que indica o termo final. Às características obtidas para cada termo é associado um peso. Cada característica é ainda composta por um conjunto de propriedades do documento, nomeadamente: marcações para o formato do termo (se está escrito apenas com caracteres minúsculos/maiúsculos ou se o primeiro caracter é maiúsculo e os seguintes minúsculos); número de termos à esquerda ou direita da entidade a extrair que se podem considerar características; classificação do termo anterior e actual.

O modelo gerado é constituído pelas características e seu peso a partir da extracção do texto anotado, que é guardado posteriormente como um objecto Java (*Annotator*) e que permite a realização de testes nos documentos aos quais se pretende extrair informação.

Depois de analisadas as ferramentas propostas para a realização de EI de documentos, no capítulo seguinte é apresentado um estudo de caso que engloba o uso destas em dois domínios distintos na área dos anúncios de venda.

Capítulo 4

Estudo de Caso

Neste capítulo será realizado um estudo de caso a um problema de Extracção de Informação mais precisamente na área dos anúncios.

Os anúncios tal como os conhecemos, apresentam uma forma desorganizada e não estruturada Informação sobre um dado produto/serviço que se pretende anunciar.

No âmbito deste estudo de caso foram utilizados para teste e avaliação de resultados anúncios de dois domínios distintos que são os seguintes: anúncios de venda de automóveis; anúncios de venda de casas.

Assim, na secção 1 são apresentados os vários conjuntos de documentos de cada domínio utilizados no estudo. Na secção 2 definem-se as várias entidades utilizadas por cada domínio bem como a estrutura das bases de dados que as suportam. Seguidamente, na secção 3, é realizado o estudo sobre o domínio dos automóveis que se divide em três fases distintas: na primeira fase é utilizada a aplicação ExtrAuto; na segunda fase a aplicação MinorThird; na terceira fase são comparados os resultados das duas soluções anteriores. Finalmente, na secção 4 é estudado o domínio das casas no qual se realizam estudos com dois CD e a aplicação MinorThird de modo a aferir quais os melhores algoritmos na extracção de cada entidade.

4.1 Conjunto de Documentos

Para a realização do conjunto de testes e avaliação de resultados, foram criados dois conjuntos de documentos, um para cada domínio (automóveis e casas).

A tarefa de preparação dos documentos dividiu-se em várias fases:

1. Recolha dos documentos na Internet

   Na fase de recolha de documentos, foi usado um programa denominado PageNest, disponível em `www.pagenest.com` e que tem como principal função a recolha completa de sítios na Internet para visualização no computador e sem recurso a uma ligação à Internet.

   Foi utilizado para a recolha de anúncios o sítio da Internet `www.slando.pt`, onde se colocam pequenos anúncios de particulares ou empresas sobre os mais variados temas. Para o trabalho em questão extraíram-se as categorias de anúncios de venda de automóveis e anúncios de venda de casas com recurso ao programa descrito acima.

2. Preparação dos documentos recolhidos

   Na preparação dos documentos recolhidos foi usado um pequeno programa em Java (parser HTML) para extrair de cada anúncio o texto correspondente ao anúncio em si, retirando todos os elementos supérfluos existentes no código fonte da página (HTML). Cada um dos anúncios extraídos foi colocado num documento de texto e colocado em pastas de acordo com o domínio que representa.

   Seguidamente analisaram-se os documentos recolhidos e foram retirados todos os documentos que não estavam escritos em língua Portuguesa bem como os documentos que não pertenciam aos domínios em estudo mas que estavam colocados, de uma forma errada, nessas categorias no sítio da Internet dos anúncios.

   No total foram obtidos 1552 anúncios de casas e 2449 anúncios de automóveis.

3. Criação dos conjuntos de documentos

   De modo a criar um conjunto de documentos diversificado utilizou-se um pequeno programa em Java para escolher aleatoriamente do conjunto inicial um conjunto menor que representa os anúncios que vão ser estudados e analisados.

   No domínio dos anúncios de casas foi criado, um conjunto inicial com 150 anúncios e um segundo conjunto com 300 anúncios que corresponde aos 150 do primeiro conjunto mais 150 novos anúncios.

   No domínio dos anúncios de automóveis foi criado apenas um conjunto com 250 anúncios que posteriormente foi duplicado para utilização no programa MinorThird

Depois da criação dos conjuntos de documentos, um documento apresenta o seguinte formato:

```
Apartamento T1 Totalmente Equipado (€49 000,00 ) Localização:
Seixal. Data: terça 8 dezembro 2009. N: 16765693 De:
967054208 contactar anunciante Disponível a partir de:
2009/12/08 Apartamento T1 totalmente equipado (mobiliário e
electrodomésticos), tudo praticamente novo, sala com kitchnette,
chão mosaico e grande roupeiro encastrado, varanda, quarto com
chão flutuante, duas janelas, garagem (parqueamento), freguesia
da Arrentela, próximo da estação comboio do Fogueteiro. Imóvel
com elevadores - 4°. andar - gas canalizado, tv cabo,pronto a
habitar. Vendo por 49.000 €
```

Nos conjuntos analisados pelo MinorThird foi criado o formato de dados reconhecido pelo programa e que consiste no uso de etiquetas XML para cada uma das entidades usadas.

Cada uma destas etiquetas tem um nome único para a entidade que identifica

e permite que o programa reconheça essas entidades e as classifique durante a sua execução.

Dos vários conjuntos de documentos criados realizaram-se anotações nos dois conjuntos de anúncios de casas e em um dos conjuntos de anúncios de automóveis.

Um documento etiquetado no formato requerido apresenta a seguinte representação:

```
<tipo_de_casa>Apartamento</tipo_de_casa> <tamanho>t2</tamanho>
em <cond_fechado>condominio fechado</cond_fechado> (€<preco>130
000,00</preco> ) Localização: <localizacao>s.felix da
marinha</localizacao>. Data: quarta 25 novembro 2009.
N: 16310623 De: contactar anunciante Disponível a partir
de: <disponivel_em>2009/11/24</disponivel_em> Vendo
<tipo_de_casa>apartamento</tipo_de_casa> <tamanho>T2</tamanho>
em excelente <cond_fechado>condominio fechado</cond_fechado>
com amplos jardins. Com <wc_equipamento>banheira
hidromassagem</wc_equipamento>, tecto com focos,
<garagem>garagem</garagem> privada.
```

Depois de apresentados os conjuntos de documentos a usar serão descritas no próximo subcapítulo quais as entidades utilizadas em cada um dos domínios.

4.2 Entidades

Durante o processo de Extracção de Informação é necessário efectuar a escolha das entidades que se pretendem extrair. Neste caso específico escolheram-se várias entidades, de acordo com o domínio em causa e do método de Extracção de Informação utilizado.

É também necessário criar o sistema de Base de Dados, que vai receber os da-

dos depois de extraídos e que representa a fase final da tarefa de Extracção de Informação.

De notar que o sistema de Base de Dados demonstrado é um sistema genérico e não um sistema especifico. A representação da Base de Dados serve como meio de suporte à escolha das entidades.

4.2.1 Domínio dos Automóveis

Para melhor entender quais as entidades utilizadas para representar este domínio é apresentada na figura 4.1 a estrutura da Base de Dados para os anúncios de automóveis.

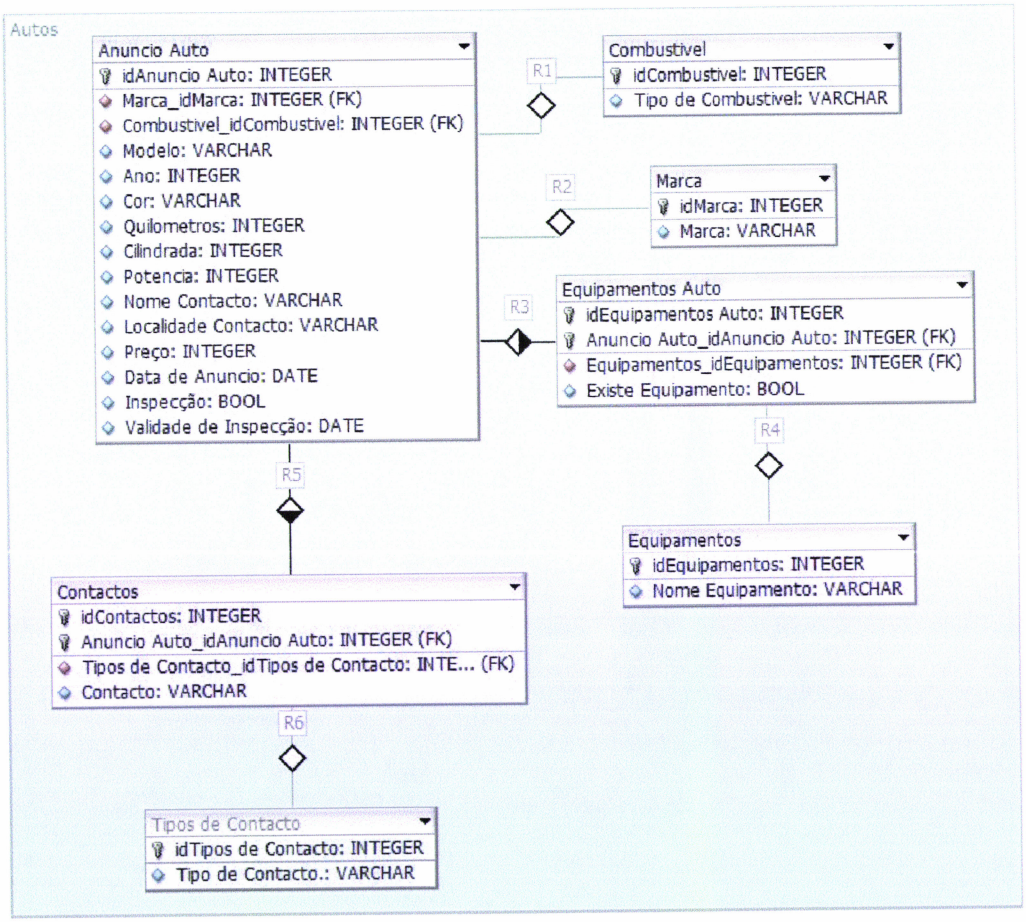

Figura 4.1: Base de Dados do domínio dos anúncios de automóveis

As entidades escolhidas dividem-se em dois grandes grupos de acordo com a técnica utilizada:

1. Os anúncios processados com o ExtrAuto tem um conjunto composto por 62 entidades.
2. Nos anúncios analisados pelo MinorThird estão definidas 6 entidades.

O primeiro conjunto de entidades (ExtrAuto) é composto pelos campos das seguintes tabelas:

- Tabela "Anuncio Auto"
  - Id, Marca, Modelo, Ano, Data do Anuncio, Preço, Localização, Cor, N.º Quilómetros, N.º Lugares, N.º Portas, Cilindrada, Potência, Inspecção Periódica Obrigatória (IPO), Data do IPO, Nome, Combustível;
- Tabela "Contactos"
  - Telefone, Telemóvel, Email, Sitio da Internet;
- Tabela "Equipamentos"
  - ABS, ESP, Vidros Eléctricos, Ar Condicionado, Rádio, Jantes de Liga Leve, Fecho Centralizado, Direcção Assistida, Espelhos Eléctricos, Alarme, Bancos Reguláveis em Altura, Volante Regulável em Altura, Airbag, Cruise Control, Sensores de Estacionamento, Sistema de Navegação, Bancos de Pele, Faróis de Nevoeiro, Bancos Aquecidos, Imobilizador, Espelhos Aquecidos, Vidros Escuros, Bancos Eléctricos, Faróis Xénon, Tecto de Abrir, Bancos Desportivos, Pintura Metalizada, Faróis com Regulação em Altura, Sensores de Chuva, Computador de Bordo, Spoiler Traseiro, Bancos Traseiros Rebatíveis, Barras no Tejadilho, Sensor de Luminosidade, Suspensão Desportiva, Capota Eléctrica (Cabrio), Encostos de

Cabeça Traseiros, Volante Desportivo, Volante com Regulação em Profundidade, Apoio de Braço, Sistema Isofix, Lava Faróis, Relógio, Conta Rotações;

O segundo conjunto de entidades é composto por elementos que representam, de uma forma geral, entidades presentes nas tabelas anteriores e que são escolhidos para a análise comparativa de Extracção de Informação entre os dois programas (ExtrAuto e MinorThird).

Assim, as entidades utilizadas estão representadas na tabela 4.1.

Entidade	Representação XML da Entidade
Marca	<marca>...</marca>
N.º Lugares	<nr_lugares>...</nr_lugares>
Vidros Eléctricos	<vidros_electricos>...</vidros_electricos>
Bancos Rebatíveis	<bancos_rebativeis>...</bancos_rebativeis>
Pintura Metalizada	<pintura_metalizada>...</pintura_metalizada>
Potência	<potencia>...</potencia>

Tabela 4.1: Entidades do domínio dos automóveis para o programa MinorThird.

4.2.2 Domínio das Casas

Tal como no caso anterior, foi criada uma Base de Dados para o domínio das casas e que é apresentada na figura 4.2.

Este domínio apresenta um conjunto de entidades definidas para o programa MinorThird e é composto por 18 entidades seleccionadas a partir das tabelas definidas na Base de Dados.

Deste modo, o conjunto de entidades é composto por entidades das seguintes tabelas:

- Tabela "Anuncio CasaC"

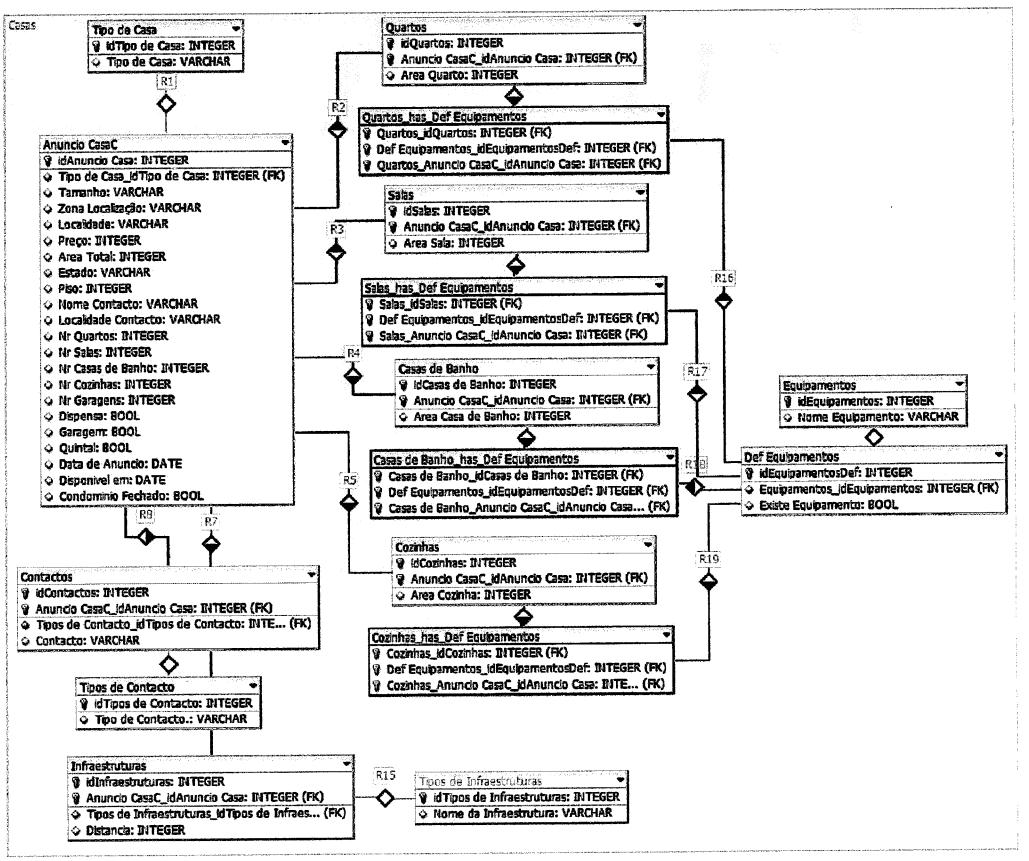

Figura 4.2: Base de Dados do domínio dos anúncios de casas

  - Tipo de Casa, Preço, Disponível em, Tamanho, Piso, Área Total, Estado, Localização, Condomínio Fechado, Garagem

- Tabela "Quartos", "Salas", "Cozinhas"e "Casas de Banho"

  - Área Quartos, Quartos Equipamentos, Área Salas, Salas Equipamentos, Área Cozinhas, Cozinhas Equipamentos, Área Casas de Banho, Casas de Banho Equipamentos

Para a representação no programa MinorThird são utilizadas, para as entidades definidas anteriormente, as seguintes anotações XML:

Entidade	Representação XML da Entidade
Tipo de Casa	<tipo_de_casa>...</tipo_de_casa>
Preço	<preco>...</preco>
Disponível em	<disponivel_em>...</disponivel_em>
Tamanho	<tamanho>...</tamanho>
Piso	<piso>...</piso>
Área Total	<area_total>...</area_total>
Estado	<estado>...</estado>
Localização	<localizacao>...</localizacao>
Condomínio Fechado	<cond_fechado>...</cond_fechado>
Garagem	<garagem>...</garagem>
Área Quartos	<quartos_area>...</quartos_area>
Quartos Equipamentos	<quartos_equipamento>...</quartos_equipamento>
Área Cozinhas	<cozinha_area>...</cozinha_area>
Cozinhas Equipamentos	<cozinha_equipamento>...</cozinha_equipamento>
Área Salas	<salas_area>...</salas_area>
Salas Equipamentos	<salas_equipamento>...</salas_equipamento>
Área Casas de Banho	<wc_area>...</wc_area>
Casas de Banho Equipamentos	<wc_equipamento>...</wc_equipamento>

Tabela 4.2: Entidades do domínio das casas para o programa MinorThird.

4.3 Extracção de Informação em Anúncios de Automóveis

O primeiro domínio a ser estudado é o dos anúncios dos automóveis no qual se apresentam os resultados da aplicação de duas ferramentas sobre o CD escolhido. A primeira fase do estudo corresponde à extracção utilizando a ferramenta ExtrAuto (baseada num sistema de regras). A segunda fase é definida pela extracção de um conjunto de entidades escolhidas utilizando a ferramenta MinorThird (utiliza algoritmos de aprendizagem automática). Por fim, na terceira fase é efectuado um estudo comparativo entre as entidades presentes nos estudos das duas fases anteriores.

4.3.1 Extracção com o programa ExtrAuto

Nesta fase é analisado o nível de extracção do programa ExtrAuto através de um conjunto de testes realizados e de medidas calculadas.

Para se realizar a análise dos dados foi criada uma folha de cálculo com os dados extraídos pelo programa na qual adicionaram-se os dados que contém as anotações sobre os dados do conjunto. Seguidamente, procedeu-se à comparação/contagem dos resultados obtidos através de um conjunto de regras.

As regras de comparação/contagem são definidas por números de um a cinco em que a cada um dos valores é atribuído o nome correspondente. Assim, ao número um que corresponde a uma extracção correcta é dado o nome de verdadeiro positivo (C de acordo com as medidas de avaliação definidas anteriormente). O número dois é descrito como uma extracção incompleta (S nas medidas de avaliação). Ao número três é utilizada a descrição de extracção incorrecta e dado o nome de falso negativo (D nas medidas de avaliação). O número quatro corresponde à ausência de valores nos dois elementos. Por fim, o número cinco é descrito como uma extracção incorrecta mas diferente da três e designa-se por falso positivo (definido por I nas medidas de avaliação).

Depois de criadas as regras realizaram-se os cálculos das medidas de precisão e de abrangência que levaram posteriormente à aplicação da fórmula de cálculo da medida F.

Estes processos foram utilizados em todas as entidades do programa ExtrAuto de modo a aferir qual a capacidade de extracção desta ferramenta.

Para uma melhor apresentação dos resultados criaram-se dois grupos de entidades com as seguintes designações: grupo das informações (contém as entidades relativas à informação do veiculo e do anunciante); grupo dos equipamentos (contém as entidades dos equipamentos do veiculo).

A tabela 4.3 contém os resultados para o grupo das informações apresentando na sua estrutura a entidade em estudo e o número de elementos pertencentes a cada

uma das regras definidas acima.

Entidade	R1	R2	R3	R4	R5
Id	250	0	0	0	0
Marca	246	1	3	0	0
Modelo	247	0	2	0	1
Ano	238	0	3	8	1
Data do Anuncio	250	0	0	0	0
Preço	247	0	3	0	0
Localização	250	0	0	0	0
Cor	231	0	2	17	0
N° de Quilómetros	216	6	4	23	1
N° de Lugares	33	0	0	216	1
N° de Portas	37	0	0	213	0
Cilindrada	165	2	11	71	1
Potência	163	0	6	77	4
IPO	35	8	0	206	1
Nome	17	1	3	228	1
Contactos	191	0	3	55	1
Combustível	245	0	5	0	0

Tabela 4.3: Contagens do grupo de informações dos anúncios automóveis

Na tabela 4.4 mostram-se os resultados dos cálculos efectuados, nomeadamente, a precisão, abrangência e medida F.

Para ilustrar os resultados obtidos é apresentada na figura 4.3 o gráfico correspondente aos dados da tabela 4.4 que na qual se comparam os vários valores obtidos, com destaque para a medida F.

De notar que os resultados deste grupo foram bastante satisfatórios, pois foram obtidos valores para a medida F superiores a 95% na grande maioria das entidades presentes no estudo (14 em 16 entidades) o que demonstra a eficiência do programa ExtrAuto nesta tarefa de Extracção de Informação.

Na segunda fase são analisadas as contagens e resultados para o grupo dos equipamentos. Este grupo é mais extenso que o anterior devido ao elevado número de equipamentos disponíveis nos automóveis modernos e ao facto de ser atribuída a cada equipamento uma entidade.

Entidade	Abrangência	Precisão	Medida F
Id	1,0000	1,0000	1,0000
Marca	0,9840	0,9960	0,9899
Modelo	0,9920	0,9960	0,9940
Ano	0,9876	0,9958	0,9917
Data do Anuncio	1,0000	1,0000	1,0000
Preço	0,9880	1,0000	0,9940
Localização	1,0000	1,0000	1,0000
Cor	0,9914	1,0000	0,9957
Nº de Quilómetros	0,9558	0,9686	0,9621
Nº de Lugares	1,0000	0,9706	0,9851
Nº de Portas	1,0000	1,0000	1,0000
Cilindrada	0,9270	0,9821	0,9538
Potência	0,9645	0,9760	0,9702
IPO	0,8140	0,7955	0,8046
Nome	0,8095	0,8947	0,8500
Contactos	0,9845	0,9948	0,9896
Combustível	0,9800	1,0000	0,9899

Tabela 4.4: Resultados do grupo de informações dos anúncios automóveis

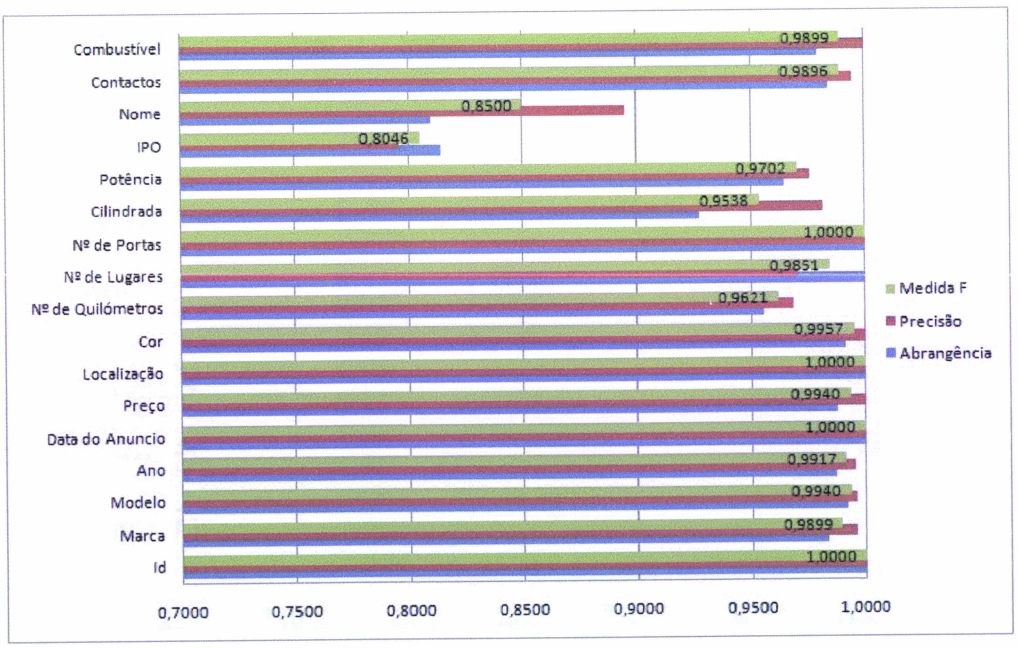

Figura 4.3: Gráfico dos resultados do grupo de informações dos anúncios automóveis

A tabela 4.5 mostra a contagem dos elementos pertencentes a cada uma das regras. Nesta tabela verifica-se a existência de um menor valor de entidades associadas à regra número um. Isso deve-se ao facto de nem todos os anúncios conterem informação sobre os equipamentos (alguns apresentam apenas informação sobre o veiculo e vendedor).

Seguidamente à tabela de contagens apresenta-se a tabela 4.6 que contém os resultados das várias medidas calculadas. A figura 4.4 apresenta o gráfico que representa os dados presentes na tabela 4.6 com principal destaque para a medida F.

Pela análise dos dados presentes no gráfico verifica-se que a maioria das entidades apresenta resultados para a medida F acima dos 95% (32 em 44 entidades). Das entidades restantes, 9 apresentam resultados entre 80% e 95%. Apenas 3 entidades tem resultados abaixo dos 80%, mas superiores a 65%.

Existem ainda 18 entidades que apresentam 100% na medida F, ou seja, foi atingida a capacidade máxima de extracção em aproximadamente 41% das entidades.

Os resultados obtidos são bastante satisfatórios na medida em que a maioria das entidades apresenta elevadas taxas de extracção. A menor taxa de extracção nalgumas entidades pode ser entendida por problemas de ambiguidade que levam o programa a assumir a entidade errada.

Depois de analisada o nível de extracção do programa ExtrAuto é tratada no próximo subcapítulo a extracção com recurso ao programa Minorthird, que utiliza métodos de aprendizagem automática para treinar o conjunto de documentos.

4.3.2 Extracção com o programa MinorThird

Depois de realizada a extracção com o programa ExtrAuto é realizado um conjunto de testes com o programa MinorThird.

Para a realização deste teste foram escolhidas algumas entidades do domínio

Entidade	R1	R2	R3	R4	R5
ABS	63	0	0	187	0
ESP	33	0	1	216	0
Vidros Eléctricos	84	0	8	158	0
Ar Condicionado	92	0	1	155	2
Rádio	82	0	1	167	0
Jantes de Liga Leve	93	0	1	156	0
Fecho Centralizado	73	0	2	175	0
Direcção Assistida	73	0	3	166	8
Espelhos Eléctricos	41	0	3	206	0
Alarme	39	0	0	211	0
Bancos Reguláveis em Altura	2	0	2	246	0
Volante Regulável em Altura	6	0	0	244	0
Airbag	59	0	3	188	0
Cruise Control	29	0	0	221	0
Sensores de Estacionamento	21	0	1	228	0
Sistema de Navegação	24	0	0	225	1
Bancos de Pele	29	0	0	220	1
Faróis de Nevoeiro	40	0	0	210	0
Bancos Aquecidos	11	0	2	237	0
Imobilizador	23	0	0	227	0
Espelhos Aquecidos	5	0	5	240	0
Vidros Escuros	5	0	0	245	0
Bancos Eléctricos	4	0	4	242	0
Faróis Xénon	20	0	0	230	0
Tecto de Abrir	26	0	1	221	2
Bancos Desportivos	6	0	3	241	0
Pintura Metalizada	49	0	3	195	3
Faróis com Reg. em Altura	5	0	1	244	0
Sensores de Chuva	8	0	0	242	0
Computador de Bordo	31	0	1	218	0
Spoiler Traseiro	3	0	0	247	0
Bancos Traseiros Rebativeis	23	0	2	220	5
Barras no Tejadilho	7	0	0	242	1
Sensor de Luminosidade	6	0	0	240	4
Suspensão Desportiva	6	0	0	244	0
Capota Eléctrica (Cabrio)	1	0	0	249	0
Encostos de Cabeça Traseiros	8	0	0	242	0
Volante Desportivo	5	0	0	245	0
Volante Reg. Profundidade	4	0	0	245	1
Apoio de Braço	11	0	1	238	0
Sistema Isofix	1	0	0	249	0
Lava Faróis	1	0	0	249	0
Relógio	1	0	0	249	0
Conta Rotações	1	0	0	249	0

Tabela 4.5: Contagens do grupo de equipamentos dos anúncios automóveis

Entidade	Abrangência	Precisão	Medida F
ABS	1,0000	1,0000	1,0000
ESP	0,9706	1,0000	0,9851
Vidros Eléctricos	0,9130	1,0000	0,9545
Ar Condicionado	0,9892	0,9787	0,9840
Rádio	0,9880	1,0000	0,9939
Jantes de Liga Leve	0,9894	1,0000	0,9947
Fecho Centralizado	0,9733	1,0000	0,9865
Direcção Assistida	0,9605	0,9012	0,9299
Espelhos Eléctricos	0,9318	1,0000	0,9647
Alarme	1,0000	1,0000	1,0000
Bancos Reguláveis em Altura	0,5000	1,0000	0,6667
Volante Regulável em Altura	1,0000	1,0000	1,0000
Airbag	0,9516	1,0000	0,9752
Cruise Control	1,0000	1,0000	1,0000
Sensores de Estacionamento	0,9545	1,0000	0,9767
Sistema de Navegação	1,0000	0,9600	0,9796
Bancos de Pele	1,0000	0,9667	0,9831
Faróis de Nevoeiro	1,0000	1,0000	1,0000
Bancos Aquecidos	0,8462	1,0000	0,9167
Imobilizador	1,0000	1,0000	1,0000
Espelhos Aquecidos	0,5000	1,0000	0,6667
Vidros Escuros	1,0000	1,0000	1,0000
Bancos Eléctricos	0,5000	1,0000	0,6667
Faróis Xénon	1,0000	1,0000	1,0000
Tecto de Abrir	0,9630	0,9286	0,9455
Bancos Desportivos	0,6667	1,0000	0,8000
Pintura Metalizada	0,9423	0,9423	0,9423
Faróis com Reg. em Altura	0,8333	1,0000	0,9091
Sensores de Chuva	1,0000	1,0000	1,0000
Computador de Bordo	0,9688	1,0000	0,9841
Spoiler Traseiro	1,0000	1,0000	1,0000
Bancos Traseiros Rebativeis	0,9200	0,8214	0,8679
Barras no Tejadilho	1,0000	0,8750	0,9333
Sensor de Luminosidade	1,0000	0,6000	0,7500
Suspensão Desportiva	1,0000	1,0000	1,0000
Capota Eléctrica (Cabrio)	1,0000	1,0000	1,0000
Encostos de Cabeça Traseiros	1,0000	1,0000	1,0000
Volante Desportivo	1,0000	1,0000	1,0000
Volante Reg. Profundidade	1,0000	0,8000	0,8889
Apoio de Braço	0,9167	1,0000	0,9565
Sistema Isofix	1,0000	1,0000	1,0000
Lava Faróis	1,0000	1,0000	1,0000
Relógio	1,0000	1,0000	1,0000
Conta Rotações	1,0000	1,0000	1,0000

Tabela 4.6: Resultados do grupo de equipamentos dos anúncios automóveis

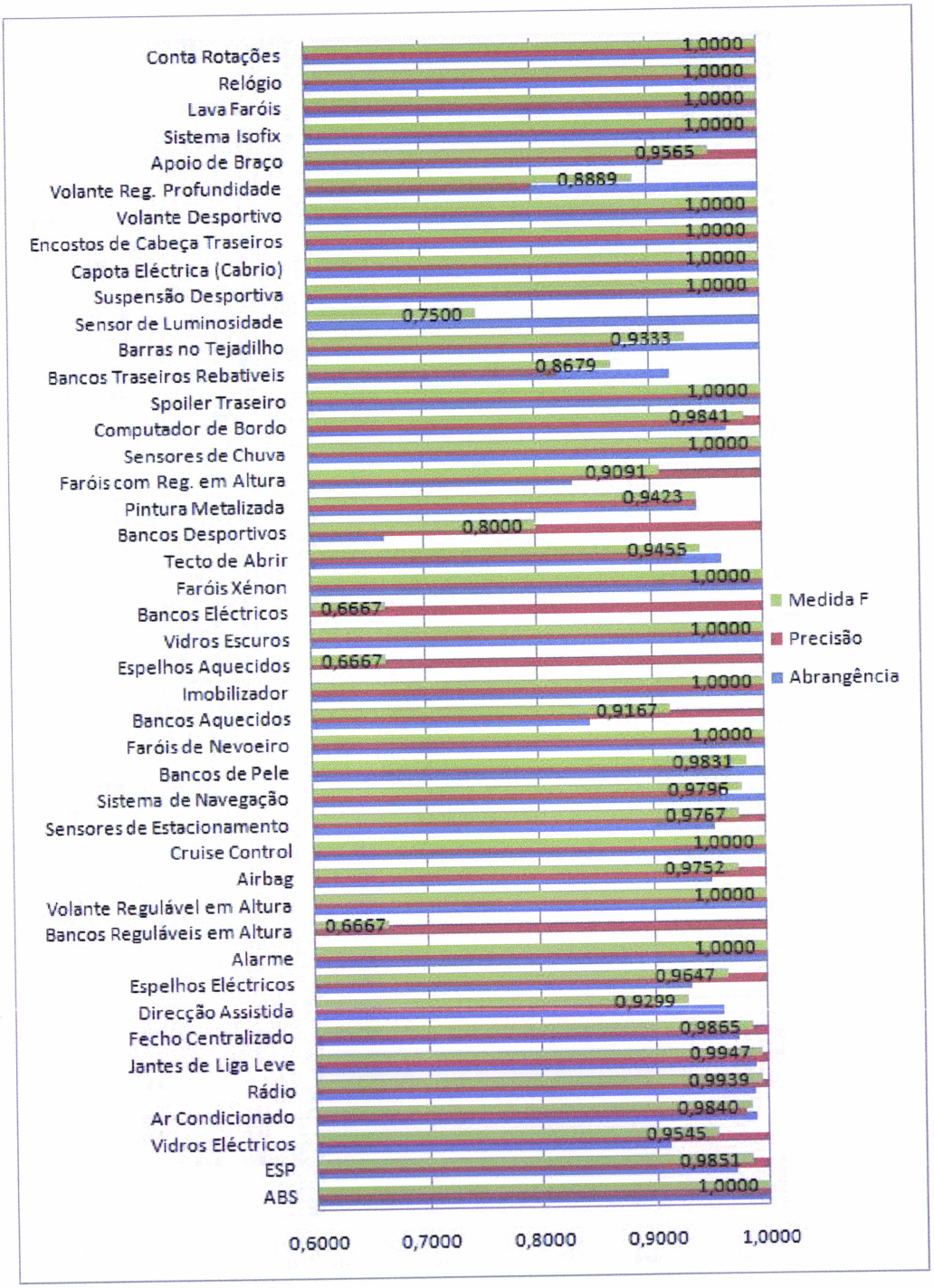

Figura 4.4: Gráfico dos resultados do grupo de equipamentos dos anúncios automóveis

dos automóveis. Deste modo foram seleccionadas seis entidades de acordo com as seguintes regras:

- Três entidades de cada grupo (três do grupo das informações + três do grupo dos equipamentos);
- As entidades de cada grupo foram escolhidas de acordo com o número de referências a essa entidade no conjunto de documentos, ou seja, como se pode ver na tabela 4.7 as entidades escolhidas de cada grupo são: para o grupo das informações, uma entidade presente em poucos documentos (n.º de lugares), uma com uma representação acima da média (potência) e uma com representação em todos os documentos (marca); no grupo dos equipamentos o número de referências nos documentos é menor mas o critério de escolha é o mesmo, ou seja, as entidades definidas são os bancos traseiros rebatíveis (30 referências), a pintura metalizada (55 referências) e os vidros eléctricos (92 referências);

Entidade	**N.º Documentos**
Marca	250
N.º Lugares	34
Potência	173
Vidros Eléctricos	92
Bancos Rebatíveis	30
Pintura Metalizada	55

Tabela 4.7: Número de documentos que contém uma entidade

Para realizar este teste foram escolhidos uma série de algoritmos de aprendizagem presentes no programa MinorThird. A escolha foi feita de acordo com as características de cada algoritmo, de modo a serem algoritmos diferentes entre si, ou no caso do VPSMM Learner e VPSMM Learner 2 que utilizam o mesmo algoritmo mas tem implementações distintas. O VPSMM Learner utiliza mais iterações, mas consome menos memória e leva mais tempo a concluir, enquanto o VPSMM

Learner 2 utiliza uma iteração apenas mas tem maior consumo de memória e menor tempo de execução.

Deste modo, os algoritmos utilizados são os seguintes:

- VPHMM Learner (Voted Perceptron Hidden Markov Model)
- VPCMM Learner (Voted Perceptron Conditional Markov Model)
- CRF Annotator Learner (Conditional Random Fields)
- SVMCMM Learner (Support Vector Machine Conditional Markov Model)
- VPSMM Learner (Voted Perceptron Hidden Semi-Markov Model)
- VPSMM Learner 2 (Voted Perceptron Hidden Semi-Markov Model, implementação 2)
- MEMM Learner (Maximum Entropy Markov Model)

De notar que foram utilizadas todas as opções predefinidas nos algoritmos implementados no programa MinorThird.

Na realização deste teste foram ainda utilizados os seguintes parâmetros: Utilização de validação cruzada com 5 pastas como método de estimação de resultados; Teste com o conjunto de documentos de modo a aferir qual o valor máximo de extracção para a entidade em cada algoritmo.

Neste primeiro teste é analisada a entidade marca. Na tabela 4.8 estão apresentados os resultados da medida F para cada um dos algoritmos a teste para os dois métodos usados. Os resultados são apresentados no gráfico 4.5 do qual se retiram as seguintes ilações:

- Todos os algoritmos apresentam uma taxa de extracção superior a 85%. Como esta entidade está presente em todos os documentos é de esperar que a taxa obtida seja elevada em todos os algoritmos.

Algoritmo	Validação C. (5 Pastas)	Pasta CD
VPHMM Learner	0,8733	0,9333
VPCMM Learner	0,9107	0,9851
CRF Annotator Learner	0,9245	0,9984
SVMCMMLearner	0,9452	0,9951
VPSMMLearner	0,9162	0,9959
VPSMMLearner2	0,8676	0,9286
MEMMLearner	0,8905	0,9681

Tabela 4.8: Resultados da medida F para a entidade marca

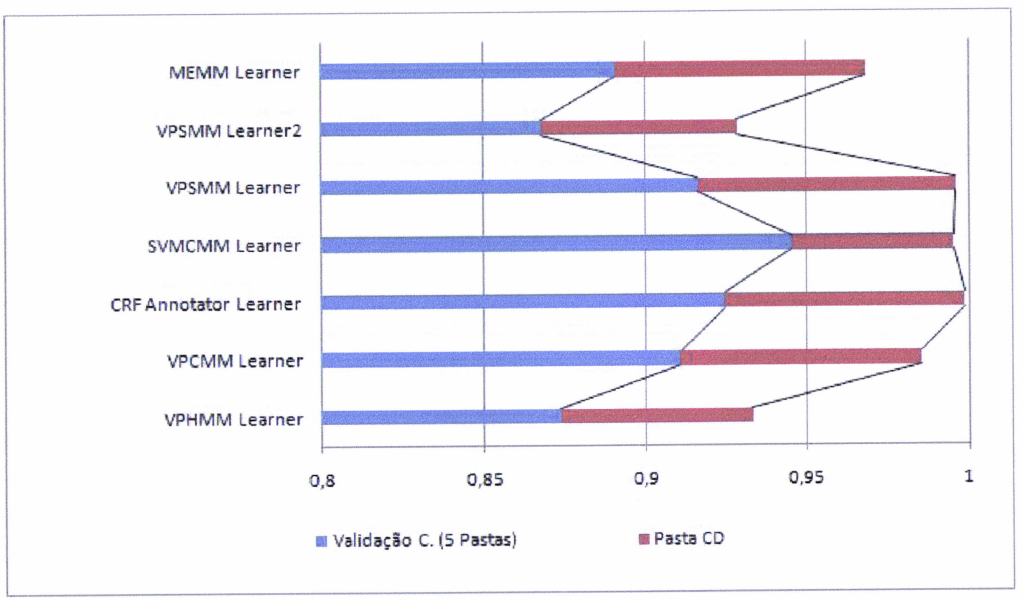

Figura 4.5: Gráfico dos resultados do programa MinorThird para a entidade marca

- O algoritmo SVMCMM é o que apresenta melhores resultados no teste de validação cruzada com 94,52%, seguido do CRF com 92,45% e do VPSMM com 91.62%.

- Quanto à comparação entre as duas implementações do VPSMM, a implementação 1 (mais iterações) apresenta melhores resultados, 91.62% contra 86,76% da implementação 2.

- No que diz respeito às diferenças entre os dois métodos (validação cruzada e pasta CD) são da ordem de 7% aproximadamente.

A próxima entidade do teste é o N.º de Lugares. Esta entidade apresenta referências em 34 documentos (o menor do grupo de informações). Na tabela 4.9 apresentam-se os valores da medida F para os métodos utilizados e no gráfico 4.6 visualizam-se esses resultados, dos quais se retiram as seguintes conclusões:

Algoritmo	**Validação C. (5 Pastas)**	**Pasta CD**
VPHMM Learner	0,0000	0,0513
VPCMM Learner	0,2609	0,6429
CRF Annotator Learner	0,5263	1,0000
SVMCMMLearner	0,7246	0,9870
VPSMMLearner	0,6154	1,0000
VPSMMLearner2	0,4231	0,7742
MEMMLearner	0,2979	0,7541

Tabela 4.9: Resultados da medida F para a entidade número de lugares

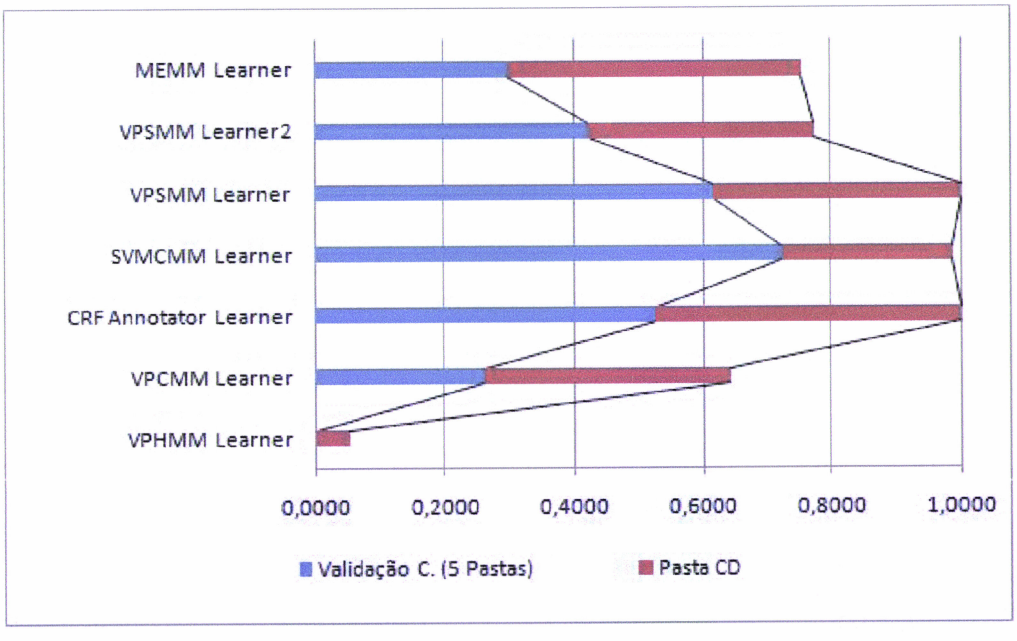

Figura 4.6: Gráfico dos resultados do programa MinorThird para a entidade número de lugares

- Os resultados obtidos são medianos, com oscilações entre 0% para o VPHMM e 72.46% para o SVMCMM.

- O algoritmo com melhor resultado é o SVMCMM com 72.46% seguido do VPSMM com 61,54% e do CRF com 52,63%.
- No algoritmo VPSMM a implementação 1 continua a obter melhores resultados 61,54% contra 42,31%.
- As diferenças entre os dois métodos são da ordem dos 34% aproximadamente.

A última entidade do grupo das informações a ser analisada é a potência. Os valores relativos aos cálculos da medida F são apresentados na tabela 4.10 e no gráfico 4.7 que ilustra esses resultados cujas conclusões são apresentadas abaixo:

Algoritmo	**Validação C. (5 Pastas)**	**Pasta CD**
VPHMM Learner	0,8378	0,9052
VPCMM Learner	0,8802	0,9474
CRF Annotator Learner	0,9443	1,0000
SVMCMMLearner	0,9461	1,0000
VPSMMLearner	0,9358	0,9979
VPSMMLearner2	0,9075	0,9556
MEMMLearner	0,8869	0,9386

Tabela 4.10: Resultados da medida F para a entidade potência

- É atingida uma taxa de extracção elevada para todos os algoritmos utilizados, acima dos 83%.
- O algoritmo que apresenta melhor resultado é o SVMCMM (94,61%) seguido do CRF(94,43%) e do VPSMM (93,58%). A diferença entre estes três algoritmos é aproximadamente de 1%.
- O algoritmo VPSMM implementação 1 obtém melhores resultados que a implementação 2.
- As diferenças entre os dois métodos situam-se nos 6% aproximadamente.

Depois de analisados todos os elementos do grupo de informações são tratados os resultados do grupo dos equipamentos. A primeira entidade a ser estudada é a

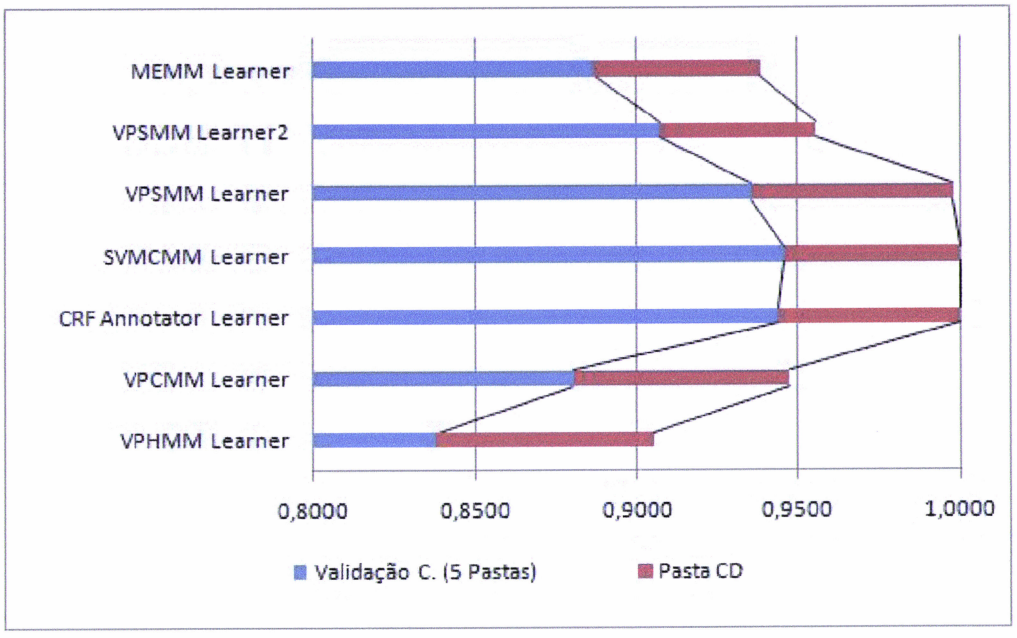

Figura 4.7: Gráfico dos resultados do programa MinorThird para a entidade potência

entidade vidros eléctricos que está representada em 92 documentos. Os resultados obtidos são apresentados na tabela 4.11 e gráfico 4.8. São evidenciadas as seguintes conclusões:

Algoritmo	**Validação C. (5 Pastas)**	**Pasta CD**
VPHMM Learner	0,7432	0,7432
VPCMM Learner	0,7347	0,8253
CRF Annotator Learner	0,9147	1,0000
SVMCMMLearner	0,9167	1,0000
VPSMMLearner	0,8605	0,9531
VPSMMLearner2	0,7231	0,9037
MEMMLearner	0,0890	0,0956

Tabela 4.11: Resultados da medida F para a entidade vidros eléctricos

- São obtidos resultados acima dos 72% para todos os algoritmos à excepção do MEMM que apresenta 8,9%.

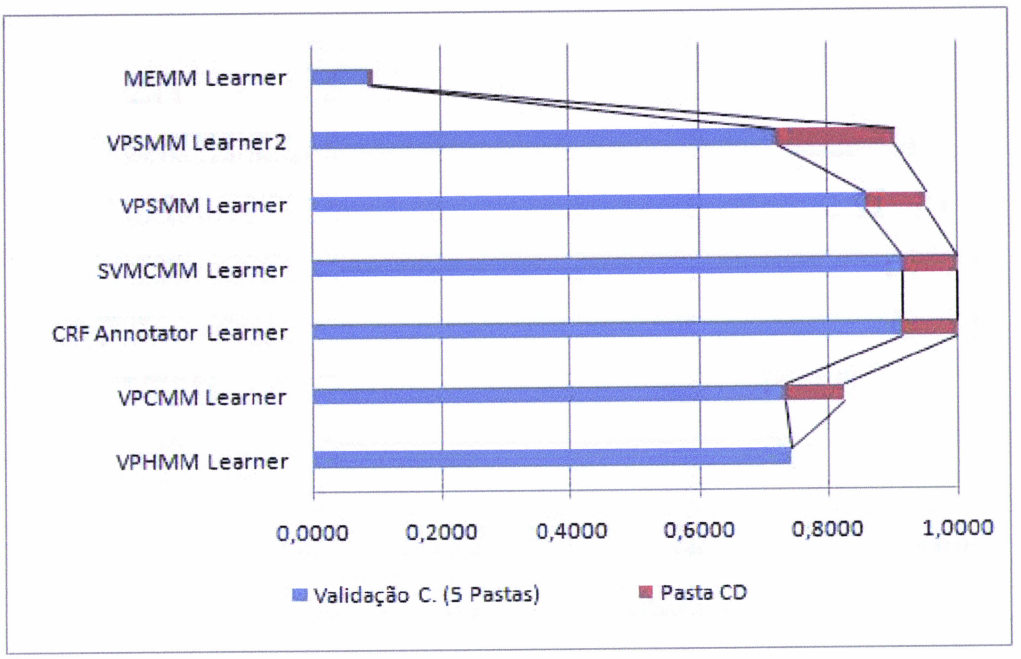

Figura 4.8: Gráfico dos resultados do programa MinorThird para a entidade vidros eléctricos

- As melhores taxas de extracção são obtidas pelos algoritmos SVMCMM (91,67%), CRF (91,47%) e VPSMM (86,05%) respectivamente.
- A implementação 1 do VPSMM apresenta uma taxa de 86,05% contra 72,31% da implementação 2.
- As diferenças entre os dois métodos em estudo situam-se aproximadamente em 8%.

A entidade bancos rebatíveis é a que apresenta menos referências no conjunto de documentos, apenas 30. Os valores da medida F são ilustrados pelo gráfico 4.9 e pela tabela 4.12 cujas ilações são as seguintes:

- A taxa de extracção é media baixa, inferior a 50% na maioria dos algoritmos, excepto nos algoritmos CRF (90,29%), SVMCMM (88,15%) e VPCMM(78,65%).

Algoritmo	Validação C. (5 Pastas)	Pasta CD
VPHMM Learner	0,3385	0,5600
VPCMM Learner	0,7865	0,9515
CRF Annotator Learner	0,9029	1,0000
SVMCMMLearner	0,8815	1,0000
VPSMMLearner	0,4675	0,5974
VPSMMLearner2	0,0000	0,0000
MEMMLearner	0,3125	0,3511

Tabela 4.12: Resultados da medida F para a entidade bancos rebatíveis

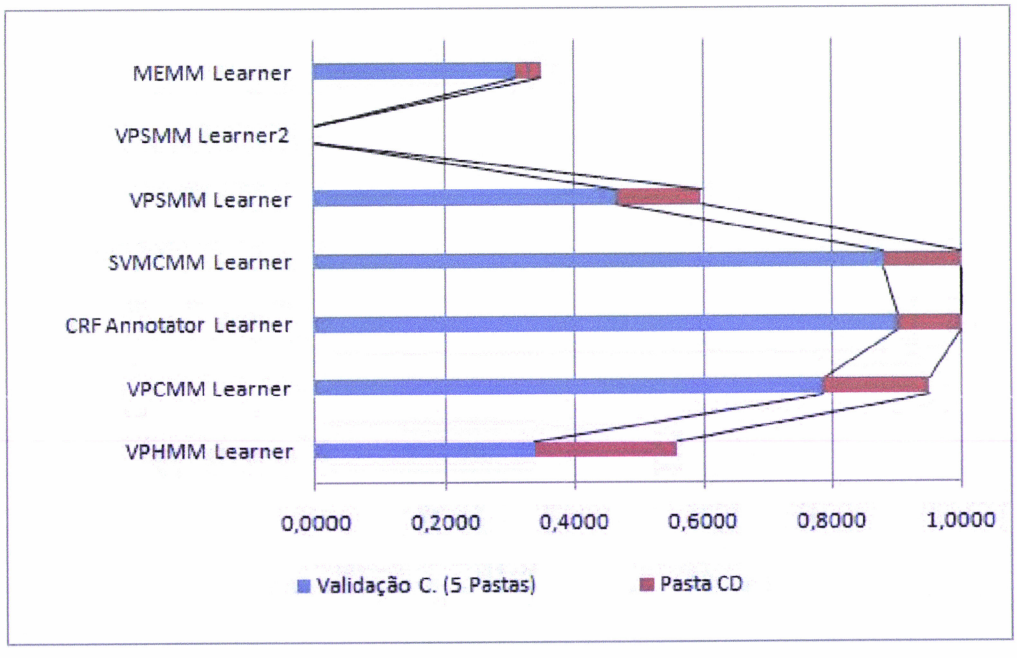

Figura 4.9: Gráfico dos resultados do programa MinorThird para a entidade bancos rebatíveis

- Na comparação entre as implementações dos algoritmos VPSMM, a implementação 1 apresenta 46,75% enquanto a implementação 2 não obtém qualquer valor.
- As diferenças entre os dois métodos cifram-se nos 11%.

A última entidade do grupo equipamentos é a pintura metalizada. Esta entidade está presente em 55 documentos do conjunto. Os resultados da medida F obtidos são apresentados no gráfico 4.10 e na tabela 4.13. Da análise dos dados obtém-se as seguintes conclusões:

Algoritmo	**Validação C. (5 Pastas)**	**Pasta CD**
VPHMM Learner	0,3303	0,4483
VPCMM Learner	0,6901	0,7143
CRF Annotator Learner	0,7260	0,9944
SVMCMMLearner	0,8554	0,9944
VPSMMLearner	0,7848	0,9944
VPSMMLearner2	0,3670	0,6047
MEMMLearner	0,4000	0,5000

Tabela 4.13: Resultados da medida F para a entidade pintura metalizada

- A taxa de extracção obtida é superior a 69% para os quatro melhores algoritmos e na ordem dos 40% para os restantes.
- Com 85,54%, o algoritmo SVMCMM é o que apresenta melhores resultados. Seguem-se o algoritmo VPSMM com 78,48% e o CRF com 72,6%.
- Mais uma vez a implementação 1 do algoritmo VPSMM obtém melhores resultados, 78,48% contra 36,7% da implementação 2.
- Os dois métodos em estudo têm uma diferença de 16% aproximadamente.

Depois de analisadas cada uma das entidades no Minorthird são observadas as seguintes conclusões acerca da extracção de informação no domínio dos automóveis:

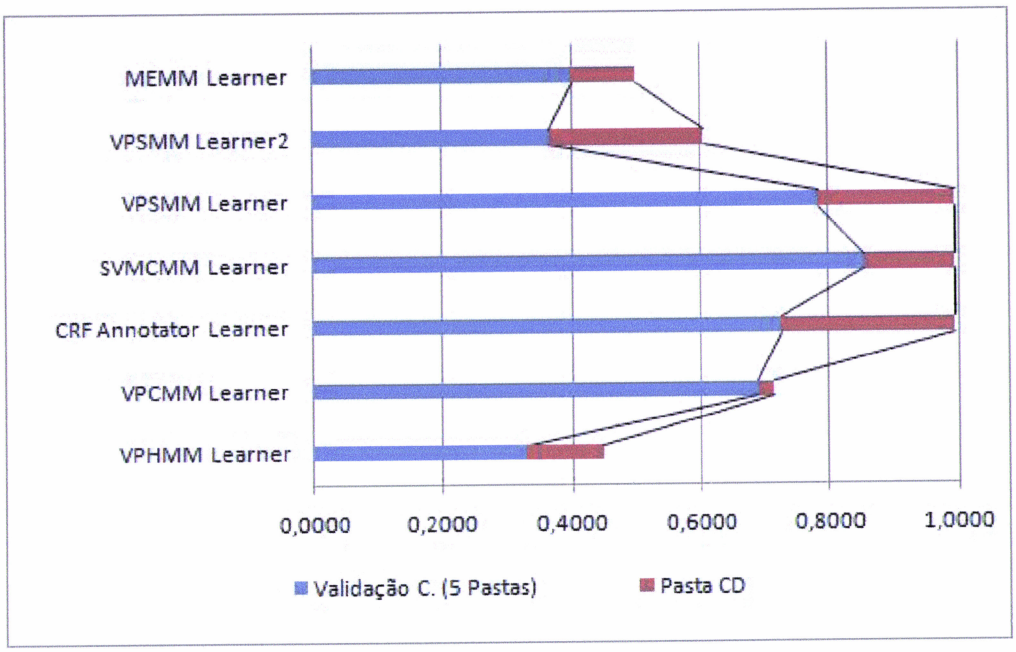

Figura 4.10: Gráfico dos resultados do programa MinorThird para a entidade pintura metalizada

- Os três algoritmos que apresentam maiores taxas de extracção são o SVMCMM, o CRF e o VPSMM. Pelo contrário o algoritmo com piores resultados é o VPHMM que necessita de uma maior quantidade de documentos de treino do que os outros algoritmos.

- O algoritmo SVMCMM apresenta resultados superiores a 72% para todas as entidades em estudo e na maioria das entidades superiores a 85%, ou seja, é o algoritmo com melhores resultados do teste.

- Na comparação directa entre as duas implementações do algoritmo VPSMM, a implementação 1 (mais tempo, menos memória) obtém sempre melhores resultados que a implementação 2 (mais memória, menos tempo) em todas as entidades no estudo.

Seguidamente à fase de análise de extracção dos programas ExtrAuto e Minorthird realiza-se no próximo subcapítulo um estudo comparativo sobre os resultados obtidos.

4.3.3 Comparação de resultados (ExtrAuto vs MinorThird)

Depois de analisados cada um dos programas de extracção nos capítulos anteriores é realizado um estudo comparativo entre os dois sistemas.

Neste estudo são utilizadas as medidas F calculadas anteriormente para cada um dos programas da seguinte forma: No ExtraAuto é utilizada a medida calculada para cada uma das entidades; no MinorThird é utilizada a medida F calculada no teste de validação cruzada com 5 pastas.

Os resultados obtidos nos testes anteriores estão presentes na tabela 4.14. De notar que no estudo com o MinotThird apenas foi utilizada a medida F do algoritmo com maior taxa de cada entidade.

Entidade	**MinorThird**		**ExtrAuto**
	Algoritmo	Medida F	Medida F
Marca	SVMCMMLearner	0,9452	0,9899
N.º de Lugares	SVMCMMLearner	0,7246	0,9851
Vidros Eléctricos	SVMCMMLearner	0,9167	0,9545
Bancos Rebatíveis	CRF Annotator Learner	0,9029	0,8679
Pintura Metalizada	SVMCMMLearner	0,8554	0,9423
Potência	SVMCMMLearner	0,9461	0,9702

Tabela 4.14: Resultados da extracção dos programas ExtrAuto e MinorThird

Os dados apresentados da tabela 4.14 estão representados no gráfico 4.11 que ilustra a comparação dos dois programas e do qual se retiram as seguintes conclusões:

- O programa ExtrAuto apresenta uma medida F superior ao MinorThird em cinco das seis entidades utilizadas.
- No programa ExtrAuto a medida F é superior a 85% em todas as entidades.
- No MinorThird as entidades que apresentam piores resultados são consequentemente as com menor presença no conjunto de documentos, ou seja, as entidades n.º de lugares, bancos rebatíveis e pintura metalizada. No entanto todas as entidades apresentam valores para a medida F superiores a 72%.

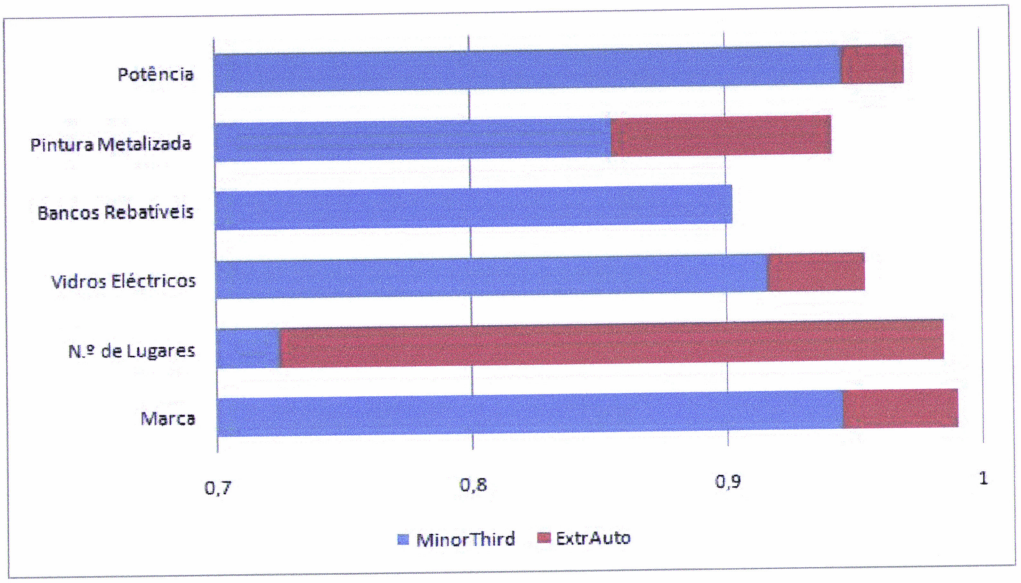

Figura 4.11: Gráfico da comparação de resultados dos programas ExtrAuto e MinorThird

- A diferença média entre os dois sistemas, com vantagem para o ExtrAuto, situa-se nos 8% aproximadamente.

Assim termina este subcapítulo no qual são analisadas as duas soluções (de regras e de aprendizagem automática) para a Extracção de Informação no domínio dos automóveis. No próximo subcapítulo é efectuado o estudo sobre o domínio das casas no qual é utilizado um sistema de aprendizagem automática e conjuntos de documentos de diferentes dimensões.

4.4 Extracção de Informação em Anúncios de Casas

Depois de concluído o conjunto de testes no domínio dos automóveis é apresentada uma nova fase de testes que incidem sobre um novo domínio, mais precisamente, o domínio dos anúncios de venda de casas.

Na realização deste conjunto de testes foi utilizado o programa MinorThird de modo a aferir qual o melhor algoritmo de aprendizagem automática a usar e quais

as relações entre o número de elementos a treinar e os valores da medida F obtidos.

Neste estudo foram utilizadas as dezoito entidades definidas anteriormente para este domínio e os seguintes algoritmos de extracção:

- VPHMM Learner (Voted Perceptron Hidden Markov Model)
- VPCMM Learner (Voted Perceptron Conditional Markov Model)
- CRF Annotator Learner (Conditional Random Fields)
- SVMCMM Learner (Support Vector Machine Conditional Markov Model)
- VPSMM Learner (Voted Perceptron Hidden Semi-Markov Model)
- VPSMM Learner 2 (Voted Perceptron Hidden Semi-Markov Model, implementação 2)
- MEMM Learner (Maximum Entropy Markov Model)

De notar que o critério de escolha foi baseado na variedade de algoritmos de modo a comparar todas as soluções conhecidas e que no caso dos algoritmos VPSMM Learner e VPSMM Learner 2 o algoritmo de aprendizagem é semelhante, no entanto, as implementações de cada um são distintas: o VPSMM utiliza menos memória mas tem um maior tempo de execução, enquanto o VPSMM 2 utiliza uma quantidade de memória mais elevada mas tem um tempo de execução menor devido ao facto de efectuar uma iteração apenas. Para cada algoritmo testado são utilizadas as condições predefinidas do programa MinorThird.

Quanto aos dois conjuntos de documentos em estudo, o primeiro conjunto é composto por 150 documentos e o segundo conjunto por 300 documentos dos quais metade são do primeiro conjunto.

Na realização deste teste foram usados os seguintes parâmetros para cada um dos conjuntos de documentos a estudo: no primeiro conjunto de documentos foi utilizada validação cruzada com 5 pastas como método de estimação de resultados;

no segundo conjunto de documentos foi utilizada validação cruzada com 10 pastas. A escolha destes valores incide no facto de o número de documentos a teste no intervalo ser igual nos dois conjuntos, ou seja, em 150 documentos por 5 pastas, são atribuídos 30 documentos por intervalo, o mesmo valor para 300 documentos em 10 pastas.

Durante o estudo foi atribuído aos resultados dos algoritmos da tarefa de extracção do programa o valor mínimo de 0,001 na medida F para os resultados cujo valor obtido foi zero. Esta medida serve para que seja possível calcular o valor da variação relativa cujo cálculo não era possível por causa da divisão por zero.

Para melhor analisar os resultados obtidos realizou-se a contagem do número de ocorrências e o número de documentos para cada entidade presente no estudo. Os resultados dessas contagens estão presentes na tabela 4.15 para os dois conjuntos de documentos utilizados.

4.4.1 Análise de resultados de cada entidade

A primeira entidade a ser analisada é a área total cujos resultados para cada algoritmo testado estão representados na tabela 4.16 através da medida F para os dois conjuntos de documentos e das variações absoluta e relativa e a partir da qual se retiram as seguintes conclusões:

- A entidade apresenta 48 ocorrências em 38 documentos do CD(150) e 90 ocorrências em 88 documentos para o CD(300). Esta entidade é essencialmente representada por caracteres numéricos que a descrevem.
- No CD(150) os algoritmos com melhores resultados são o SVMCMM com 62,56% seguido do CRF com 47,47% e do VPSMM com 45,33%, ao passo que, no CD(300) a lista é composta pelo CRF com 73,85% seguido do SVMCMM com 71,75% e do VPSMM com 61,33%.

Entidade	CD (150)		CD (300)	
	Refs.	Nº Docs.	Refs.	Nº Docs.
Área Cozinhas	19	19	46	44
Tipo de Casa	199	118	389	236
Área WC's	21	12	37	23
Preço	176	150	351	299
Disponível em	150	150	300	300
Quartos Equipamentos	137	73	220	123
Área Quartos	43	20	94	41
Área Total	40	38	90	88
Cozinhas Equipamentos	162	38	250	68
Tamanho	232	134	452	270
Piso	30	24	46	37
Área Salas	41	31	75	62
Garagem	74	60	147	122
Salas Equipamentos	96	56	194	122
Estado	98	74	192	143
Localização	300	149	615	299
Condomínio Fechado	13	10	29	21
WC's Equipamentos	57	35	92	62

Tabela 4.15: Resultados das contagens de ocorrências e número de documentos para cada entidade.

Algoritmo	CD (150)	CD (300)	Variação	Variação
	Medida F	Medida F	Absoluta	Relativa (%)
VPHMM Learner	0,0001	0,1447	0,1446	144600,00
VPCMM Learner	0,0001	0,0956	0,0955	95500,00
CRF Annotator Learner	0,4747	0,7385	0,2638	55,57
SVMCMMLearner	0,6256	0,7175	0,0919	14,69
VPSMMLearner	0,4533	0,6133	0,1600	35,30
VPSMMLearner2	0,0001	0,1765	0,1764	176400,00
MEMMLearner	0,0001	0,0001	0,0000	0,00

Tabela 4.16: Resultados do programa MinorThird para a entidade área total

- A maior variação absoluta pertence ao algoritmo CRF que obtém um aumento de 0,2638 na medida F enquanto o algoritmo SVMCMM é um dos que regista menor aumento (0,0919). Quanto à variação relativa, nestes dois algoritmos situa-se nos 55,57% e 14,69% respectivamente.

- Relativamente aos algoritmos com pior performance, o destaque vai para o MEMM que não apresenta qualquer valor para os dois CD e para o VPCMM, VPHMM e VPSMM2 que apenas apresentam resultados para o CD(300).

- Quanto às duas implementações do algoritmo VPSMM, a implementação um obteve melhores resultados nos dois CD.

- Os resultados obtidos são um pouco díspares pois apenas os três algoritmos com melhores resultados apresentam uma taxa de extracção satisfatória enquanto os restantes apenas apresentam resultados residuais ou nulos.

A segunda entidade a ser analisada é o condomínio fechado e cujos resultados estão apresentados na tabela 4.17 e a partir dos quais se retiram as seguintes ilações:

Algoritmo	**CD (150)**	**CD (300)**	**Variação**	**Variação**
	Medida F	**Medida F**	**Absoluta**	**Relativa (%)**
VPHMM Learner	0,0001	0,0241	0,0240	24000,00
VPCMM Learner	0,4348	0,4231	-0,0117	-2,69
CRF Annotator Learner	0,3636	0,8435	0,4799	131,99
SVMCMMLearner	0,4286	0,828	0,3994	93,19
VPSMMLearner	0,4348	0,7785	0,3437	79,05
VPSMMLearner2	0,0001	0,1778	0,1777	177700,00
MEMMLearner	0,0001	0,1364	0,1363	136300,00

Tabela 4.17: Resultados do programa MinorThird para a entidade condomínio fechado

- Esta entidade é a que está menos representada nos conjuntos de documentos, ou seja, no CD(150) apenas apresenta 13 ocorrências em 10 documentos e no CD(300) 29 em 21 documentos. Esta entidade é representada pelas palavras que identificam a casa como pertencente a um condomínio fechado.

- Os algoritmos com melhores resultados para o CD(150) são os seguintes: VPCMM (43,48%), VPSMM(43,48%) e SVMCMM(42,86%). Para o CD(300) a melhor performance pertence ao CRF com 84,35% seguido do SVMCMM (82,80%) e do VPSMM com 77,85%.

- Quanto às variações entre os dois CD, os três algoritmos com melhores resultados apresentam aumentos absolutos de 0,4799, 0,3994 e 0,3437, respectivamente. Esta variação transmite-se numa variação relativa da ordem dos 80% para o algoritmo VPSMM chegando aos 132% para o CRF. De notar que o algoritmo com melhor resultado no CD(150) (VPCMM) apresenta uma variação negativa, ainda que quase nula.

- Os piores resultados são atribuídos ao VPHMM, MEMM e VPSMM2 que não apresentam valores para o CD(150) e que no CD(300) apresentam valores residuais(VPHMM) ou inferiores a 20%.

- A implementação um do algoritmo VPSMM apresenta melhores resultados do que a segunda implementação, ou seja, 77,85% contra 17,78%.

- Apesar desta entidade ter pouca representação no CD os resultados obtidos são bastante satisfatórios para os três melhores algoritmos que atingem taxas superiores a 77%. No entanto, para os restantes os resultados são muito diminutos.

A próxima entidade é a área da cozinha da qual são apresentados os resultados obtidos na tabela 4.18 e evidenciam-se as seguintes conclusões:

- A entidade está pouco representada no CD, ou seja, são obtidas 19 ocorrências em 19 documentos para o CD(150) e 46 em 44 documentos para o CD(300). Esta entidade é representada por caracteres numéricos que identificam o elemento que se pretende extrair.

- No CD(150) os algoritmos com melhores resultados são o SVMCMM com 66,04% , o CRF com 51,68% e o VPSMM com 18,95%, enquanto, no CD(300)

Algoritmo	CD (150)	CD (300)	Variação	Variação
	Medida F	Medida F	Absoluta	Relativa (%)
VPHMM Learner	0,0001	0,0897	0,0896	89600,00
VPCMM Learner	0,0882	0,4084	0,3202	363,04
CRF Annotator Learner	0,5169	0,7705	0,2536	49,06
SVMCMMLearner	0,6604	0,8509	0,1905	28,85
VPSMMLearner	0,1895	0,5656	0,3761	198,47
VPSMMLearner2	0,0001	0,0001	0,0000	0,00
MEMMLearner	0,0001	0,1019	0,1018	101800,00

Tabela 4.18: Resultados do programa MinorThird para a entidade área da cozinha

os melhores continuam a ser os mesmos algoritmos na mesma ordem mas com valores de 85,09%, 77,05% e 56,56%, respectivamente.

- Nesta entidade a maior variação absoluta é obtida no algoritmo VPSMM com 0,3761 seguido do VPCMM com 0,3202. No que diz respeito aos algoritmos com melhores resultados, a variação absoluta é de 0,1905 para o SVMCMM e de 0,2536 para o CRF. As variações relativas para os três melhores algoritmos são de 28,85%, 49,06 e 198,47% (SVMCMM, CRF e VPSMM).

- O algoritmo com piores resultados é o VPSMM2 que apresenta valores nulos para os dois CD. Os algoritmos VPHMM e MEMM também obtém valores muito baixos (inferiores a 10%).

- Devido ao facto da implementação dois do VPSMM apresentar valores nulos, verifica-se que a implementação um volta a obter melhores resultados.

- Os resultados obtidos são satisfatórios para os dois melhores algoritmos (SVMCMM e CRF) que atingem taxas superiores a 77%. No entanto para os restantes algoritmos os resultados não são suficientes, apesar de no VPSMM e VPCMM se obterem resultados na ordem dos 50%, os algoritmos MEMM, VPHMM e VPSMM 2 apresentam resultados diminutos.

Os equipamentos existentes numa cozinha são a próxima entidade a ser analisada. Os cálculos realizados são apresentados na tabela 4.19 e evidenciam as seguintes

conclusões:

Algoritmo	CD (150) Medida F	CD (300) Medida F	Variação Absoluta	Variação Relativa (%)
VPHMM Learner	0,0099	0,1551	0,1452	1466,67
VPCMM Learner	0,3436	0,3398	-0,0038	-1,11
CRF Annotator Learner	0,7139	0,7913	0,0774	10,84
SVMCMMLearner	0,7963	0,75	-0,0463	-5,81
VPSMMLearner	0,4194	0,515	0,0956	22,79
VPSMMLearner2	0,1212	0,1456	0,0244	20,13
MEMMLearner	0,0533	0,037	-0,0163	-30,58

Tabela 4.19: Resultados do programa MinorThird para a entidade equipamento da cozinha

- Esta entidade apresenta uma boa quantidade de ocorrências para os dois conjuntos de documentos(162 no CD(150) e 250 no CD(300)) e está presente aproximadamente em 1/5 dos documentos de cada CD. Esta entidade é essencialmente definida pelas palavras que definem os vários equipamentos existentes numa cozinha.

- Quanto aos algoritmos com melhores resultados observa-se o seguinte: no CD(150) o SVMCMM obtém 79,63%, o CRF 71,39% e o VPSMM 41,94%; no CD(300) os algoritmos com melhores resultados continuam a ser os mesmos mas o CRF obtém melhor resultado que o SVMCMM (79,13% contra 75%) e o VPSMM continua no terceiro lugar com 51,5%.

- A maior variação absoluta pertence ao VPHMM com 0,1452 enquanto todos os outros apresentam variações inferiores a 0,1. Dos três algoritmos com melhores resultados a maior variação relativa foi obtida no VPSMM com 22,79% ao passo que o SVMCMM apresenta uma variação negativa de 5,81%.

- O algoritmo com pior performance é o MEMM que para além de apresentar uma variação negativa os seus resultados são inferiores a 6%. Também os algoritmos VPHMM e VPSMM2 apresentam resultados diminutos.

- Na comparação entre implementações do VPSMM, a implementação um apresenta melhores resultados.
- Para os dois algoritmos com melhor performance, o CRF e SVMCMM os resultados são satisfatórios e na ordem dos 75%. Mas, para a maioria dos algoritmos os resultados são muito baixos. No VPSMM e VPCMM são da ordem dos 40% mas nos restantes são inferiores a 16%.

A entidade disponível em identifica-se como a data a partir da qual uma casa fica disponível para ser habitada. Os resultados dos cálculos para esta entidade são visualizados na tabela 4.20 e permitem tirar as seguintes ilações:

Algoritmo	CD (150) Medida F	CD (300) Medida F	Variação Absoluta	Variação Relativa (%)
VPHMM Learner	0,9696	0,9937	0,0241	2,49
VPCMM Learner	0,8318	0,9635	0,1317	15,83
CRF Annotator Learner	1	1	0,0000	0,00
SVMCMMLearner	1	1	0,0000	0,00
VPSMMLearner	0,0001	0,0001	0,0000	0,00
VPSMMLearner2	0,0001	0,0001	0,0000	0,00
MEMMLearner	0,9705	0,9499	-0,0206	-2,12

Tabela 4.20: Resultados do programa MinorThird para a entidade disponível em

- Esta entidade está representada em todos os documentos dos dois CD e apresenta um número de ocorrências semelhante ao número de documentos. É definida por palavras que descrevem a data a ser extraída.
- Devido à representação desta entidade no conjunto de documentos verifica-se que no caso do CD(150) os algoritmos com melhores resultados são o CRF e SVMCMM com 100% e o MEMM com 97,05%. No CD(300) os resultados são semelhantes aos do CD(150) mas o terceiro algoritmo com melhores resultados é o VPHMM com 99,37%.
- As variações absoluta e relativa são baixas ou nulas em todos os algoritmos excepto no VPCMM que obteve 0,1317 de variação absoluta e 15,83% de

variação relativa. O único algoritmo a obter variações negativas, apesar de quase nulas, foi o MEMM com -2,12%.

- Os piores resultados são os do algoritmo VPSMM nas suas duas implementações que apresentam resultados nulos para os dois CD. Na comparação entre as duas implementações verifica-se um empate apesar de não existir qualquer significado neste resultado.

- Os resultados obtidos nesta entidade são excelentes pois a taxa de extracção é superior a 95% em 5 dos 7 algoritmos e de 100% em 2. No entanto em 2 dos algoritmos os resultados são nulos.

A entidade estado é definida pela apresentação em que a casa se encontra e é definida pelas palavras que descrevem essa entidade. Os resultados obtidos são mostrados na tabela 4.21 e evidenciam-se as conclusões abaixo:

Algoritmo	CD (150)	CD (300)	Variação	Variação
	Medida F	Medida F	Absoluta	Relativa (%)
VPHMM Learner	0,0001	0,0962	0,0961	96100,00
VPCMM Learner	0,1796	0,336	0,1564	87,08
CRF Annotator Learner	0,493	0,6485	0,1555	31,54
SVMCMMLearner	0,6104	0,6985	0,0881	14,43
VPSMMLearner	0,5821	0,6844	0,1023	17,57
VPSMMLearner2	0,1125	0,3947	0,2822	250,84
MEMMLearner	0,0132	0,1146	0,1014	768,18

Tabela 4.21: Resultados do programa MinorThird para a entidade estado

- No CD(150) esta entidade está presente em 74 documentos em 98 ocorrências, enquanto no CD(300) os valores são de 192 e 143, respectivamente.

- Os algoritmos com melhores resultados na medida F são: para o CD(150) o SVMCMM com 61,04%, o VPSMM com 58,21% e o CRF com 49,3%; no CD(300) os algoritmos são os mesmos mas com maiores taxas de extracção 69,85%, 68,44% e 64,85% respectivamente.

- A maior variação absoluta ocorre no algoritmo VPSMM2 com um aumento de 0,2822. Nos algoritmos com melhor taxa de extracção (SVMCMM, VPSMM e CRF) a variação absoluta é de 0,0881, 0,1023 e 0,1555 o que resulta numa variação relativa de 14,43%, 17,57% e 31,54%.

- No que diz respeito ao piores resultados, o VPHMM e o MEMM são os que menor taxa de extracção apresentam (inferior a 12%). Os resultados para o VPCMM e VPSMM2 também são baixos, na ordem dos 35%.

- O algoritmo VPSMM na implementação um apresenta melhores resultados do que a segunda implementação

- Os resultados obtidos nesta entidade são satisfatórios para os três algoritmos com maior valor na medida F (na ordem dos 67%), no entanto para os restantes são observados resultados baixos, ou diminutos.

A entidade garagem define se uma casa está equipada com um ou mais lugares de estacionamento coberto e é representada pelas palavras que a identificam. Os resultados obtidos para os dois CD utilizados e as variações absoluta e relativa são apresentadas na tabela 4.22 a partir da qual se mostram as seguintes conclusões:

Algoritmo	CD (150)	CD (300)	Variação	Variação
	Medida F	Medida F	Absoluta	Relativa (%)
VPHMM Learner	0,0001	0,4322	0,4321	432100,00
VPCMM Learner	0,688	0,8132	0,1252	18,20
CRF Annotator Learner	0,8227	0,8805	0,0578	7,03
SVMCMMLearner	0,8649	0,8963	0,0314	3,63
VPSMMLearner	0,7568	0,8754	0,1186	15,67
VPSMMLearner2	0,672	0,8836	0,2116	31,49
MEMMLearner	0,5143	0,5806	0,0663	12,89

Tabela 4.22: Resultados do programa MinorThird para a entidade garagem

- Esta entidade para o CD(150) está representada em 60 documentos com 74 ocorrências e para o CD(300) em 122 documentos e 147 ocorrências.

- Para o CD(150) os melhores resultados são obtidos pelo algoritmo SVMCMM (86,49%) seguido do CRF (82,27%) e do VPSMM (75,68%). No CD(300) o com maior medida F é novamente o SVMCMM (89,63%), em segundo o VPSMM2 (88,36%) e por fim o CRF (88,05%).

- As maiores variações absolutas são do VPHMM que tem um aumento de 0,4321 e do VPSMM2 com 0,2116, o que lhe permite obter um dos melhores resultados no CD(300). Quanto aos restantes algoritmos, CRF e SVMCMM a variação absoluta é baixa (CRF → 0,0578 e SVMCMM → 0,0314) o que se reflecte numa variação relativa também baixa (CRF → 7,03% e SVMCMM → 3,63%).

- O VPHMM e o MEMM são os que apresentam piores resultados. O VPHMM apesar de ter a maior variação apenas obtém 43,22% no CD(300) enquanto o MEMM apresenta uma variação relativa de 12,89% e assim obtém 58.06% no mesmo CD.

- Nesta entidade o algoritmo VPSSM na segunda implementação obteve melhores resultados do que na primeira.

- Esta entidade apresenta resultados bastante satisfatórios pois a maioria dos algoritmos utilizados, 5 em 7, obteve resultados acima dos 81%. Nos restantes algoritmos os resultados são baixos, da ordem dos 50%.

A localização é a próxima entidade a ser analisada. Esta define-se como o local onde está situada uma casa e representa-se pelas palavras que a identificam. Os resultados obtidos pelo programa são ilustrados na tabela 4.23 a partir da qual se evidenciam as seguintes conclusões:

- Esta entidade está presente em quase todos os documentos e com um número de ocorrências superior ao número de documentos no CD. Assim, para o CD(150) está inserida em 149 documentos e ocorre 300 vezes enquanto no CD(300) esses valores são 299 ocorrências e 615 documentos.

Algoritmo	CD (150)	CD (300)	Variação	Variação
	Medida F	Medida F	Absoluta	Relativa (%)
VPHMM Learner	0,5238	0,6933	0,1695	32,36
VPCMM Learner	0,4899	0,6424	0,1525	31,13
CRF Annotator Learner	0,7591	0,8249	0,0658	8,67
SVMCMMLearner	0,7579	0,8218	0,0639	8,43
VPSMMLearner	0,5	0,5638	0,0638	12,76
VPSMMLearner2	0,392	0,5004	0,1084	27,65
MEMMLearner	0,0916	0,0838	-0,0078	-8,52

Tabela 4.23: Resultados do programa MinorThird para a entidade localização

- No que diz respeito à análise dos valores da medida F para cada um dos CD verifica-se que para o CD(150) os algoritmos CRF (75,91%), SVMCMM (75,79%) e VPHMM (52,38%) são os que apresentam resultados mais elevados. No CD(300) os algoritmos são os mesmos mas os valores obtidos são mais elevados, 82,49%, 82,18% e 69,33%, respectivamente.

- Com uma variação absoluta de 0,1695 e relativa de 32,36% o VPHMM foi o algoritmo que mais evoluiu na análise aos dois CD. Também o VPCMM e o VPSMM2 tiveram variações semelhantes ao VPHMM. Os algoritmos com melhores resultados na medida F (CRF e SVMCMM) tiveram uma variação absoluta baixa que corresponde a uma variação relativa aproximada de 8%.

- O algoritmo com pior performance nesta entidade é o MEMM que apresenta resultados residuais e obtém variação relativa negativa no teste.

- Nesta entidade a implementação um do VPSMM têm resultados mais elevados do que a segunda implementação do algoritmo.

- Os resultados obtidos para a entidade localização são satisfatórios para 2 dos 7 algoritmos do teste que apresentam resultados superiores a 80% e razoáveis para 4 dos 5 restantes com resultados na ordem dos 60%. Os pior resultado é observado no algoritmo MEMM com 8,38% apenas.

A entidade apresentada na tabela 4.24 define-se pelo piso em que a casa se insere. Esta entidade é essencialmente representada no CD por o número que a identifica ou, em alguns casos, pela palavra. De acordo com os resultados obtidos observam-se as seguintes conclusões:

Algoritmo	CD (150)	CD (300)	Variação	Variação
	Medida F	Medida F	Absoluta	Relativa (%)
VPHMM Learner	0,0001	0,0779	0,0778	77800,00
VPCMM Learner	0,1852	0,1939	0,0087	4,70
CRF Annotator Learner	0,0001	0,6394	0,6393	639300,00
SVMCMMLearner	0,6069	0,6543	0,0474	7,81
VPSMMLearner	0,5769	0,4583	-0,1186	-20,56
VPSMMLearner2	0,06	0,1852	0,1252	208,67
MEMMLearner	0,0412	0,0001	-0,0411	-99,76

Tabela 4.24: Resultados do programa MinorThird para a entidade piso

- Esta é uma entidade que está presente em poucos documentos e com poucas ocorrências no CD. Assim, no CD(150) tem 30 referências em 24 documentos e no CD(300) 46 em 37 documentos.

- Para o CD(150) os algoritmos com melhores resultados são o SVMCMM com 60,69% e o VPSMM com 57,69% enquanto os restantes apenas apresentam resultados residuais. No CD(300) o SVMCMM volta a apresentar o melhor resultado (65,43%) seguido do CRF(63,94%).

- Nesta entidade o CRF é o algoritmo que apresenta maior variação relativa, pois no CD(150) apresenta um resultado nulo enquanto no CD(300) é o segundo melhor do teste. O melhor do teste CD(300) (SVMCMM) apresenta uma variação relativa de 7,81%. De notar que o VPSMM e o MEMM sofrem variações negativas com destaque para o VPSMM que atinge -20,56% de variação relativa.

- Os piores algoritmos do teste são o MEMM e o VPHMM que apresentam resultados nulos ou residuais (MEMM → 0% VPHMM → 7,79%). Também o

VPCMM e o VPSMM2 apresentam resultados muito baixos.

- A implementação um do VPSMM apresenta melhores resultados do que a segunda variante, ou seja, 45,83% contra 18,52%.
- Os resultados obtidos para esta entidade são pouco satisfatórios na medida em que apenas 2 algoritmos obtém resultados da ordem dos 60% enquanto os restantes apresentam resultados muito baixos ou diminutos.

A entidade que se segue é a mais importante em qualquer anúncio de venda, ou seja, o preço. Na tabela 4.25 mostram-se os resultados da medida F para os vários algoritmos a teste e evidenciam-se as seguintes conclusões:

Algoritmo	**CD (150)**	**CD (300)**	**Variação**	**Variação**
	Medida F	**Medida F**	**Absoluta**	**Relativa (%)**
VPHMM Learner	0,927	0,9588	0,0318	3,43
VPCMM Learner	0,7463	0,9195	0,1732	23,21
CRF Annotator Learner	0,9378	0,9644	0,0266	2,84
SVMCMMLearner	0,9528	0,9664	0,0136	1,43
VPSMMLearner	0,9289	0,9506	0,0217	2,34
VPSMMLearner2	0,9417	0,9551	0,0134	1,42
MEMMLearner	0,3263	0,0256	-0,3007	-92,15

Tabela 4.25: Resultados do programa MinorThird para a entidade preço

- Esta entidade é apresentada no CD pelo valor numérico que a identifica e descreve. Para o CD(150) é referenciada 176 vezes nos 150 documentos do conjunto. No CD(300) os valores calculados são de 351 ocorrências em 299 documentos.
- Neste conjunto de testes os algoritmos com melhores resultados são: CD(150) SVMCMM (95,28%), VPSMM2 (94,17%) e CRF (93,78%); CD(300) SVMCMM (94,64%), CRF (96,44%) e VPHMM (95,88%).

- Todos os algoritmos apresentam variações muito baixas à excepção do VPCMM que apresenta uma variação relativa de 23,21% e do MEMM que apresenta uma variação negativa muito acentuada de 92,15%.

- O pior algoritmo deste teste é o MEMM que apresenta resultados muito baixos nos dois CD com especial destaque para o CD(300) onde obtém resultados residuais (2,56%).

- Na entidade preço a segunda implementação do VPSMM apresenta melhores resultados nos dois CD do que a primeira. No entanto os obtidos resultados são muito semelhantes para o CD(300).

- Os resultados obtidos neste teste são excelentes pois em 5 das 7 entidades são obtidos valores acima dos 95%. No entanto o algoritmo MEMM apresenta resultados residuais para esta entidade.

A entidade cujos resultados do teste estão representados na tabela 4.26 é descrita pela área que um quarto ocupa numa casa. As conclusões acerca do teste da entidade área do quarto são as seguintes:

Algoritmo	CD (150)	CD (300)	Variação	Variação
	Medida F	Medida F	Absoluta	Relativa (%)
VPHMM Learner	0,0001	0,1166	0,1165	116500,00
VPCMM Learner	0,0482	0,1617	0,1135	235,48
CRF Annotator Learner	0,4115	0,7865	0,3750	91,13
SVMCMMLearner	0,6227	0,8838	0,2611	41,93
VPSMMLearner	0,2062	0,4654	0,2592	125,70
VPSMMLearner2	0,0001	0,0001	0,0000	0,00
MEMMLearner	0,0366	0,0385	0,0019	5,19

Tabela 4.26: Resultados do programa MinorThird para a entidade área dos quartos

- A área dos quartos é descrita no CD pelo valor numérico que a identifica. Para o CD(150) ocorre 43 vezes em 20 documentos e no CD(150) 94 em 41 documentos.

- Os algoritmos com melhores resultados são: para o CD(150) o algoritmo SVMCMM com 62,27% e o CRF com 41,15%; no CD(300) os algoritmos são os mesmos mas os valores mais elevados (88,38% e 78,65%).

- Na entidade área dos quartos as maiores variações absolutas pertencem aos algoritmos CRF e SVMCMM que tiveram um aumento de 0,3750 e 0,2611 que se traduz numa variação relativa de 91,13% e 41,93%, respectivamente.

- Os piores resultados são obtidos pelo VPSMM2 (resultado nulo nos dois CD) seguido do MEMM com valores residuais e do VPHMM e VPCMM com resultados inferiores a 17%.

- Neste algoritmo a primeira implementação do VPSMM obtém melhores resultados pois na segunda variante não são observados resultados nos testes.

- Esta entidade apresenta resultados satisfatórios para os dois algoritmos com melhores resultados (SVMCMM e CRF) mas nos restantes 5 algoritmos observam-se resultados muito baixos, menores que 21%.

A tabela 4.27 apresenta os resultados obtidos para a entidade equipamento dos quartos que é representada pelas palavras que descrevem cada um dos equipamentos existentes nesta divisão da casa.

Algoritmo	**CD (150)**	**CD (300)**	**Variação**	**Variação**
	Medida F	**Medida F**	**Absoluta**	**Relativa (%)**
VPHMM Learner	0,0001	0,1058	0,1057	105700,00
VPCMM Learner	0,1839	0,281	0,0971	52,80
CRF Annotator Learner	0,5559	0,6221	0,0662	11,91
SVMCMMLearner	0,5577	0,5734	0,0157	2,82
VPSMMLearner	0,4762	0,5691	0,0929	19,51
VPSMMLearner2	0,1654	0,4733	0,3079	186,15
MEMMLearner	0,0502	0,1781	0,1279	254,78

Tabela 4.27: Resultados do programa MinorThird para a entidade equipamento dos quartos

- Com 137 ocorrências em 73 documentos para o CD(150) e 220 ocorrências em 123 documentos esta entidade apresenta-se em média quantidade nos dois CD.
- Para o CD(150) o SVMCMM com 76,15% é o algoritmo com melhores resultados. No segundo teste com o CD(300) o SVMCMM volta a apresentar o valor mais elevado (78,49%) seguido do CRF com 76,32% e do VPSMM 59,50%.
- Dos algoritmos com melhores resultados a maior variação absoluta e relativa pertence ao CRF com 39,30% de variação relativa. De notar que as variações no SVMCMM são muito reduzidas (variação relativa de 3,07%).
- No CD(150) três dos algoritmos apresentam resultados nulos (VPHMM, VPSMM2 e MEMM) enquanto no CD(300) os piores são o MEMM com um valor residual e o VPSMM2 com um resultado abaixo dos 15%.
- A implementação um do VPSMM apresenta maior valor na medida F do que a segunda implementação.
- Para esta entidade os resultados obtidos são satisfatórios para os três algoritmos com melhor performance (SVMCMM, CRF e VPSMM) mas para os restantes são baixos, inferiores a 31% e no caso do MEMM residuais.

A entidade área das salas é definida pelo número que a representa no CD. Os resultados obtidos para os dois CD em estudo são apresentados na tabela 4.28 e obtém-se as seguintes ilações:

- Esta entidade está presente em 31 documentos com 41 ocorrências para o CD(150) enquanto no CD(300) os valores são de 75 ocorrências em 62 documentos.
- No CD(150) o algoritmo com melhor resultado é o SVMCMM com 76,15% enquanto o CRF e o VPSMM apresentam valores da ordem dos 50%. Nos resultados dos testes com o CD(300) o SVMCMM volta a obter o resultado

Algoritmo	CD (150)	CD (300)	Variação	Variação
	Medida F	Medida F	Absoluta	Relativa (%)
VPHMM Learner	0,0001	0,3103	0,3102	310200,00
VPCMM Learner	0,1579	0,298	0,1401	88,73
CRF Annotator Learner	0,5479	0,7632	0,2153	39,30
SVMCMMLearner	0,7615	0,7849	0,0234	3,07
VPSMMLearner	0,4785	0,595	0,1165	24,35
VPSMMLearner2	0,0001	0,1226	0,1225	122500,00
MEMMLearner	0,0001	0,04	0,0399	39900,00

Tabela 4.28: Resultados do programa MinorThird para a entidade área das salas

mais elevado com 78,49% seguido do CRF com 76,32% e do VPSMM com 59,50%.

- A maior variação absoluta do teste para os algoritmos com maior medida F pertence ao CRF com 0,2153 a que corresponde uma variação relativa de 39,30%. O VPSMM apresenta uma variação relativa de 24,35% enquanto o SVMCMM apresenta uma variação muito baixa (3,07%).

- Os piores valores no CD(150) são atribuídos a três algoritmos que apresentam resultados nulos (VPHMM, VPSMM2 e MEMM). No CD(300) o pior resultado é atribuído ao MEMM que apresenta um valor residual e ao VPSMM2 cuja medida F é abaixo dos 15%.

- A primeira implementação do VPSMM apresenta melhores resultados do que a segunda nos dois CD.

- Os resultados obtidos são satisfatórios para os dois melhores algoritmos (SVMCMM e CRF) com valores na ordem dos 77%. Os restantes não acompanham esses resultados, apenas o VPSMM obtém 59,5% enquanto nos outros os valores são abaixo dos 32%.

Os equipamentos existentes nas salas são a próxima entidade em estudo cujos resultados estão presentes na tabela 4.29 e dos quais se retiram as seguintes conclusões:

Algoritmo	CD (150)	CD (300)	Variação	Variação
	Medida F	Medida F	Absoluta	Relativa (%)
VPHMM Learner	0,0179	0,108	0,0901	503,35
VPCMM Learner	0,4414	0,461	0,0196	4,44
CRF Annotator Learner	0,6429	0,6145	-0,0284	-4,42
SVMCMMLearner	0,6702	0,631	-0,0392	-5,85
VPSMMLearner	0,3448	0,5272	0,1824	52,90
VPSMMLearner2	0,0696	0,3551	0,2855	410,20
MEMMLearner	0,303	0,1911	-0,1119	-36,93

Tabela 4.29: Resultados do programa MinorThird para a entidade equipamento das salas

- No CD(150) existem 96 referências à entidade em 56 documentos enquanto no CD(300) esses valores são de 194 e 122, respectivamente.
- Os maiores valores de medida F obtidos são: para o CD(150), SVMCMM (67,02%) e CRF (64,29%); no CD(300) SVMCMM (63,10%) e CRF (61,45%).
- Os algoritmos com melhores resultados apresentam variações relativas negativas (ainda que pouco acentuadas) na ordem dos 5%. As maiores variações absolutas pertencem às duas implementações do algoritmo VPSMM.
- No conjunto de testes do CD(150) os piores resultados são obtidos pelo VPHMM e pelo VPSMM2 que apresentam valores residuais. Para o CD(300) o VPHMM e o MEMM são os com pior performance.
- A implementação um do VPSMM apresenta maiores valores de medida F do que a segunda implementação do algoritmo.
- Nesta entidade os resultados obtidos são pouco satisfatórios. São obtidos no máximo valores da ordem dos 60% para os dois melhores algoritmos. Dos 5 algoritmos restantes o VPSMM e o VPCMM obtém números da ordem dos 48% e os outros abaixo dos 20%.

O tamanho da casa é a próxima entidade estudada cujos resultados estão representados na tabela 4.30 e a partir dos quais se retiram as seguintes conclusões:

Algoritmo	CD (150)	CD (300)	Variação	Variação
	Medida F	Medida F	Absoluta	Relativa (%)
VPHMM Learner	0,8412	0,9053	0,0641	7,62
VPCMM Learner	0,7692	0,7991	0,0299	3,89
CRF Annotator Learner	0,9566	0,9433	-0,0133	-1,39
SVMCMMLearner	0,963	0,9382	-0,0248	-2,58
VPSMMLearner	0,8706	0,9116	0,0410	4,71
VPSMMLearner2	0,8189	0,9211	0,1022	12,48
MEMMLearner	0,3599	0,1377	-0,2222	-61,74

Tabela 4.30: Resultados do programa MinorThird para a entidade tamanho

- Esta entidade é das que mais referências obtém nos CD. Deste modo, no CD(150) está presente em 134 documentos com 232 ocorrências e no CD(300) esses valores sobem para 270 e 452 respectivamente.
- Para o CD(150) os melhores resultados pertencem ao SVMCMM (96,30%) e ao CRF (95,66%) enquanto no CD(300) o CRF com 94,33% é o algoritmo com valor mais elevado e ao qual se segue o SVMCMM e VPSMM2 com 93,82% e 92,11%.
- A maior variação relativa positiva pertence ao VPSMM2 com 12,48% enquanto os outros algoritmos com melhores resultados (SVMCMM e CRF) obtém variações negativas na ordem dos 3%.
- O MEMM é o pior algoritmo nos testes desta entidade, pois obteve no CD(150) um resultado de 35,99% que no segundo teste baixa para 13,77% o que dá uma variação relativa negativa de 61,74%.
- A segunda implementação do VPSMM apresenta melhor resultado do que a primeira versão do algoritmo.
- Os resultados obtidos para a entidade tamanho são muito bons pois 5 dos 7 algoritmos apresentam valores superiores a 90%. No entanto nota-se nos algoritmos com melhores resultados uma variação negativa que indica um número elevado de elementos de treino.

A entidade tipo de casa é definida pelas palavras que a identificam (apartamento, moradia, vivenda, etc...) e cujos resultados para os dois CD são apresentados na tabela 4.31 e a partir dos quais se retiram as seguintes ilações:

Algoritmo	CD (150)	CD (300)	Variação	Variação
	Medida F	Medida F	Absoluta	Relativa (%)
VPHMM Learner	0,7651	0,8284	0,0633	8,27
VPCMM Learner	0,9396	0,8804	-0,0592	-6,30
CRF Annotator Learner	0,9622	0,9004	-0,0618	-6,42
SVMCMMLearner	0,9509	0,9135	-0,0374	-3,93
VPSMMLearner	0,9138	0,8789	-0,0349	-3,82
VPSMMLearner2	0,9288	0,8945	-0,0343	-3,69
MEMMLearner	0,8739	0,6623	-0,2116	-24,21

Tabela 4.31: Resultados do programa MinorThird para a entidade tipo de casa

- Esta entidade é das que mais referências têm no CD(150) onde são identificadas 199 ocorrências em 118 documentos. No CD(300) existem 389 ocorrências em 236 documentos.
- Para o CD(150) os três algoritmos com melhor performance são o CRF com 96,22%, o SVMCMM com 95,09% e o VPCMM com 93,96%. No CD(300) o melhor resultado é obtido pelo SVMCMM (91,35%) seguido do CRF (90,04%) e do VPSMM2 (89,45%).
- Nesta entidade à excepção do VPHMM que apresenta uma variação relativa positiva (8,27%) todos os restantes algoritmos apresentam variações negativas com principal destaque para o MEMM que apresenta uma variação de -24,21%.
- Devido à sua prestação no CD(300) o MEMM é o algoritmo com pior resultado do teste apresentando uma medida F de 66,23%.
- A segunda implementação do VPSMM apresenta melhores resultados do que a primeira nos dois CD.
- Os resultados obtidos na entidade tipo de casa são muito bons pois foram obtidos valores superiores a 87% em 5 dos 7 algoritmos e nos restantes 2

superiores a 66%. No entanto, os resultados com o CD(150) são superiores aos do CD(300) o que indica que com um maior número de dados de treino não são obtidos melhores resultados nesta entidade.

A entidade área da casa se banho é a penúltima entidade a ser analisada e é caracterizada pelo número que a identifica no CD. Os resultados obtidos nos testes com os dois CD são mostrados na tabela 4.32 e observam-se as seguintes conclusões:

Algoritmo	**CD (150)**	**CD (300)**	**Variação**	**Variação**
	Medida F	**Medida F**	**Absoluta**	**Relativa (%)**
VPHMM Learner	0,0001	0,0001	0,0000	0,00
VPCMM Learner	0,0001	0,1818	0,1817	181700,00
CRF Annotator Learner	0,5893	0,775	0,1857	31,51
SVMCMMLearner	0,7737	0,7237	-0,0500	-6,46
VPSMMLearner	0,283	0,2667	-0,0163	-5,76
VPSMMLearner2	0,0001	0,0001	0,0000	0,00
MEMMLearner	0,0001	0,0001	0,0000	0,00

Tabela 4.32: Resultados do programa MinorThird para a entidade área das casas de banho

- A entidade é a que têm menos presenças nos dois CD, ou seja, no CD(150) existe em 12 documentos com 21 ocorrências e no CD(300) em 23 documentos e 37 ocorrências.
- Para o CD(150) o melhor algoritmo é o SVMCMM com 77,37% enquanto os restantes são obtidos valores muito baixos. No CD(300) o CRF é o que apresenta a melhor performance com 77,50% seguindo-se o SVMCMM com 72,37%.
- Nos algoritmos com melhores resultados a maior variação relativa pertence ao CRF com 31,51%. O SVMCMM apresenta uma variação negativa de 6,46%.
- Devido ao facto desta entidade estar em reduzido número nos CD, os algoritmos VPHMM, VPSMM2 e MEMM apresentam valores nulos nos dois testes.

Deste modo, a implementação um do VPSMM obteve melhor resultado do que a segunda variante.

- Os resultados obtidos para os dois algoritmos com melhor performance são satisfatórios mas para os restantes cinco não se pode referir o mesmo pois obtiveram-se valores baixos ou nulos.

A última entidade a ser analisada é os equipamentos da casa de banho. Os resultados dos testes efectuados com os dois CD são visualizados na tabela 4.33 a partir da qual se retiram as seguintes ilações:

Algoritmo	CD (150)	CD (300)	Variação	Variação
	Medida F	Medida F	Absoluta	Relativa (%)
VPHMM Learner	0,0001	0,045	0,0449	44900,00
VPCMM Learner	0,1622	0,3282	0,1660	102,34
CRF Annotator Learner	0,4778	0,6686	0,1908	39,93
SVMCMMLearner	0,5702	0,7093	0,1391	24,39
VPSMMLearner	0,5222	0,6486	0,1264	24,21
VPSMMLearner2	0,1854	0,4058	0,2204	118,88
MEMMLearner	0,0432	0,2204	0,1772	410,19

Tabela 4.33: Resultados do programa MinorThird para a entidade equipamento das casas de banho

- Esta entidade apresenta-se com 57 ocorrências em 35 documentos para o CD(150) e 92 ocorrências em 62 documentos no CD(300).
- São obtidos valores pouco satisfatórios no CD(150) no qual o SVMCMM com 57,02% é o algoritmo com melhor resultado. Para o CD(300) o SVMCMM obteve 70,93% seguido do CRF com 66,86% e do VPSMM com 64,86%.
- Nesta entidade todos os algoritmos sofrem variações positivas. O SVCMM e o CRF apresentam uma variação relativa de 24,39% e 39,93%, respectivamente.
- O VPHMM têm a pior performance dos testes pois apresenta um resultado nulo no CD(150) e um valor residual no CD(300)

- Com 64,86% a primeira implementação do VPSMM mostra melhores resultados do que a segunda implementação em ambos os CD.
- Para a entidade equipamento da casa de banho os resultados obtidos são pouco satisfatórios para os três algoritmos com melhores resultados (SVMCMM, CRF e VPSMM) e baixos para os restantes 4 algoritmos o que mostra a necessidade de uma maior presença da entidade nos CD.

Depois de efectuada a análise a cada uma das entidades definidas nos anúncios de casas é realizado na secção seguinte um estudo comparativo dos algoritmos com melhores resultados obtidos.

4.4.2 Estudo comparativo dos melhores resultados obtidos

Nesta última secção do caso de estudo é realizado um estudo comparativo entre os melhores algoritmos de cada entidade estudada de modo a aferir qual o melhor algoritmo a usar na tarefa de extracção de informação de anúncios de venda de casas.

Na tabela 4.34 mostram-se os valores da medida F para o algoritmo com o resultado mais elevado de cada entidade estudada. Para este estudo comparativo foram escolhidos os algoritmos do CD(300) pois é o que contém maior número de documentos e que de uma forma geral apresenta melhores resultados para o domínio em estudo. A última coluna da tabela é composta pelo tempo de execução, em segundos, dos algoritmos utilizados.

De modo a obter tempos de execução comparáveis entre si foi utilizada em todos os testes uma máquina com as seguintes características de hardware e software: Intel D930 3.0 Ghz com 2.5 Gb de memória Ram e disco rígido de 500 GB; Sistema Operativo Linux Kubuntu e JavaSE 6; Para executar o programa MinorThird numa tarefa de extracção é necessário aumentar a memória da máquina de Java para 1024Mb para que não ocorram erros.

Entidade	Algoritmo	Medida F	Tempo de Execução (s)
Área Total	CRF Annotator Learner	0,7385	9777
Condomínio Fechado	CRF Annotator Learner	0,8435	5904
Área Cozinha	SVMCMMLearner	0,8509	175
Equipamento Cozinha	CRF Annotator Learner	0,7913	10287
Disponível em	SVMCMMLearner	1,0000	146
Estado	SVMCMMLearner	0,6985	628
Garagem	SVMCMMLearner	0,8963	327
Localização	CRF Annotator Learner	0,8249	9387
Piso	SVMCMMLearner	0,6543	235
Preço	SVMCMMLearner	0,9664	547
Área Quarto	SVMCMMLearner	0,8838	229
Equipamento Quarto	CRF Annotator Learner	0,6221	9218
Área Sala	SVMCMMLearner	0,7849	221
Equipamento Sala	SVMCMMLearner	0,6310	774
Tamanho	CRF Annotator Learner	0,9433	10235
Tipo de Casa	SVMCMMLearner	0,9135	475
Área WC	CRF Annotator Learner	0,7750	6653
Equipamento WC	SVMCMMLearner	0,7093	377

Tabela 4.34: Algoritmos com melhor resultado para as entidades analisadas no MinorThird

Os resultados da medida F obtidos são ilustrados no gráfico 4.12 e a partir do qual se retiram as seguintes conclusões:

- A taxa mínima obtida é de 62,21% para a entidade Equipamento do Quarto e a segunda menor é para o Equipamento da Sala. De notar que todas as entidades baseadas em equipamentos das várias divisões obtém resultados abaixo dos 79,13% (Equipamento da Cozinha) apesar destas entidades terem uma boa presença no CD.
- O SVMCMM é algoritmo com melhores resultados nas várias entidades, mais precisamente em 11 das 18 estudadas. Este algoritmo é também o que apresenta um menor tempo de execução, abaixo dos 800 segundos em todas as entidades estudadas.

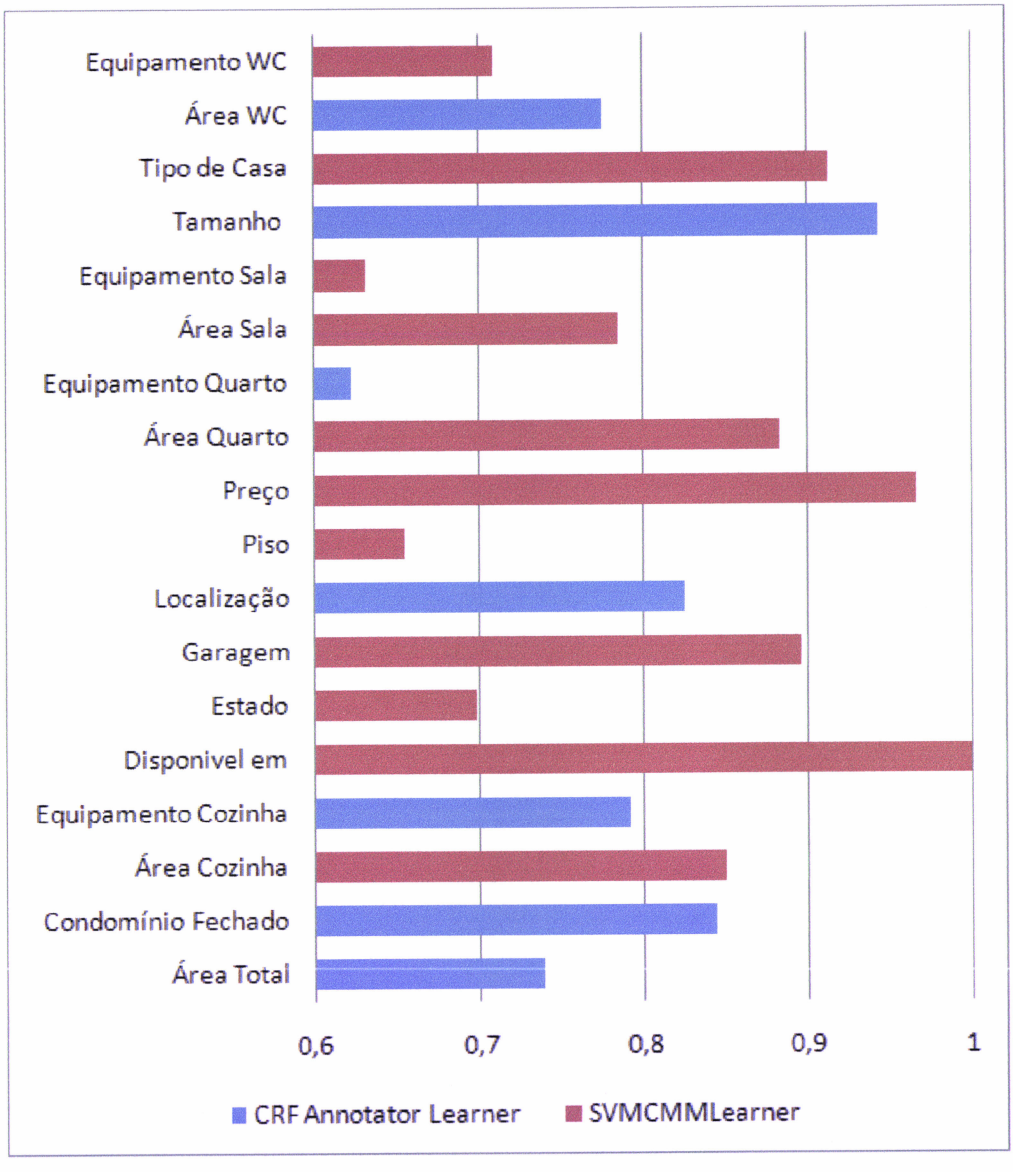

Figura 4.12: Gráfico da comparação dos melhores resultados de cada entidade

- O CRF apresenta melhores resultados em 7 das 18 entidades estudadas. No entanto o tempo de execução é bastante elevado quando comparado com o SVMCMM.

- Nas entidades em que o CRF apresenta melhores resultados do que o SVMCMM e pela análise das tabelas da secção anterior verifica-se que nestas entidades a diferença entre as medidas F dos dois algoritmos é: na Área Total é 2,1%; no Condomínio Fechado é 1,55%; no Equipamento da Cozinha é 4,13%; na Localização é 0,31%; no Equipamento dos Quartos é 4,87%; no Tamanho é de 0,51%; na Área do WC é 5,13%. No entanto as diferenças entre os tempos de execução são aproximadamente 10 vezes maiores no CRF relativamente ao SVMCMM.

- No que diz respeito à taxa de extracção das várias entidades, 4 das 18 entidades apresentam valores superiores a 90%, 5 das 18 valores entre 80% e 90%, 5 das 18 valores entre 70% e 80% e as restantes 4 valores abaixo dos 70%. Estes resultados dão origem a uma média de 80,7% e a um desvio padrão de 11,53%.

Assim termina o estudo de caso nos anúncios de venda dos domínios das casas e dos automóveis. No capítulo seguinte são analisadas as conclusões obtidas e o trabalho futuro na área da EI.

Capítulo 5

Conclusões e Trabalho Futuro

Os trabalhos apresentados nesta dissertação tiveram como objectivo o estudo da Extracção de Informação de documentos em língua Portuguesa, nomeadamente na área dos anúncios de venda de automóveis e casas. Utilizaram-se as duas grandes abordagens no que diz respeito a sistemas de Extracção de Informação, ou seja, foi desenvolvida uma ferramenta baseada no sistema de regras (ExtrAuto) para a Extracção de Informação de anúncios de venda de automóveis e foi utilizada uma ferramenta que implementa vários algoritmos de aprendizagem automática (MinorThird (Cohen, 2004a)) para a realização de testes de análise de algoritmos com vários conjuntos de documentos na área dos anúncios de venda de casas e para comparação de resultados entre si e o programa ExtrAuto nos anúncios de automóveis.

5.1 Conclusões

No que diz respeito ao domínio dos anúncios de automóveis, o desenvolvimento da ferramenta ExtrAuto permitiu a realização da tarefa de Extracção de Informação para um conjunto composto por 250 anúncios escolhidos aleatoriamente de um conjunto inicial de 2449 provenientes de uma fonte na Internet (`http://www.slando.`

pt). O programa desenvolvido é composto por um conjunto de entidades sobre o domínio em causa em que para cada uma são implementadas regras em conjunto com uma fonte de conhecimento. Este formato permite a Extracção de Informação quando no documento em análise uma entidade é identificada na base de conhecimento e consequentemente aplicada a(s) regra(s) de extracção para essa entidade. Por fim, os elementos extraídos para cada documento do conjunto são escritos numa folha de cálculo juntamente com os resultados anotados manualmente e calculadas as medidas de precisão, abrangência e medida F para cada uma das entidades.

Para o programa ExtrAuto foram definidas um total de 62 entidades divididas em dois grupos principais (informações e equipamentos) de acordo com a temática a que pertencem. Para as 16 entidades do grupo das informações foram obtidos resultados na medida F superiores a 80% em todas as entidades. Em 14 das 16 obtiveram-se resultados superiores a 95% e em 8 dessas superiores a 99%. O grupo dos equipamentos, constituído por 44 entidades, apresenta resultados superiores a 95% em 32 entidades, entre 80% e 95% para 9 entidades e entre 65% e 80% para as 3 entidades restantes. Este conjunto de resultados evidência a boa prestação da aplicação na Extracção de Informação de anúncios de venda de automóveis na medida em que a maioria apresenta resultados superiores a 95%.

Para o conjunto de testes efectuados com a ferramenta MinorThird foram escolhidas 6 entidades das 62 definidas anteriormente. Os critérios de escolha utilizados definem-se pelo grupo a que pertencem (informações e equipamentos) e pelo número de ocorrências no conjunto de documentos (baixa, média e alta). Na primeira fase o conjunto de testes visa apurar qual o algoritmo com maior valor na medida F para cada entidade. Na segunda fase é comparado esse resultado com o obtido pelo programa ExtrAuto para essa entidade. Os valores da medida F obtidos na primeira fase variam entre os 72% e 95% com 4 das 6 entidades a obterem valores acima dos 90%. Quanto aos algoritmos com melhor performance, em 5 das 6 entidades foi para o SVMCMM essa atribuição e na entidade restante para o CRF. Na comparação das medidas F entre as duas ferramentas o sistema ExtrAuto apresenta melhores

resultados em 5 das 6 entidades apresentando uma diferença média de 8% entre si e o MinorThird, ou seja, verifica-se a vantagem da ferramenta baseada no sistema de regras, relativamente ao sistema de aprendizagem automática para as entidades em estudo.

De notar, que a ferramenta ExtrAuto apenas pode ser utilizada no domínio em causa e no caso da modificação deste são necessárias alterações profundas no sistema (dicionário e regras) que pode ser uma tarefa muito morosa. A ferramenta MinorThird pode utilizar um qualquer domínio desde que se procedam às correspondentes anotações nos conjuntos de documentos de treino.

No segundo domínio em estudo, os dois conjuntos de documentos utilizados foram escolhidos aleatoriamente de uma fonte na Internet(`www.slando.pt`), composta por 1552 anúncios de venda de casas. O primeiro conjunto é composto por 150 documentos e o segundo por 300 documentos (150 anteriores + 150 novos). O conjunto de testes neste domínio é efectuado pela ferramenta MinorThird, visa verificar as alterações na medida F para cada algoritmo e qual o algoritmo com melhores performances para os conjuntos de documentos em 18 entidades escolhidas para o efeito.

Nos testes com os dois conjuntos de documentos verifica-se um aumento da medida F na grande maioria dos casos até ser atingido um ponto de saturação, ou seja, quando os elementos utilizados como treino no primeiro conjunto de documentos são em quantidade e qualidade, os resultados obtidos com esse conjunto e com o segundo conjunto são muito semelhantes existindo apenas diferenças residuais nos valores da medida F. Nos restantes casos existe uma variação absoluta positiva que é mais ou menos acentuada de acordo com a entidade em estudo e com o algoritmo utilizado.

No que diz respeito aos algoritmos com melhores e piores resultados, o SVMCMM é o que obtém melhores resultados seguido do CRF. Quanto aos valores da medida F, estes variam entre 62,21% e 100%. Os algoritmos HMM e MEMM são os que apresentam piores resultados, apresentando em alguns casos valores nulos para os dois conjuntos estudados. Relativamente a estes algoritmos verifica-se a influência da

qualidade e quantidade do conjunto de treino, na medida em que para apresentarem bons resultados necessitam que esse conjunto tenha mais elementos e de melhor qualidade do que para os algoritmos CRF e SVMCMM. No entanto, quando o conjunto é completo os resultados obtidos são muito semelhantes aos obtidos pelos algoritmos com melhor performance.

5.2 Trabalho Futuro

De modo a aumentar os valores da medida F observados, no sistema desenvolvido para a EI de anúncios de automóveis (ExtrAuto), a utilização de um método de desambiguação para os casos em que existem dificuldades na atribuição de entidades. O uso de um sistema de prioridades é uma das várias soluções disponíveis para a resolução deste problema.

O aumento do número de elementos presentes na base de conhecimento para as expressões que identificam os equipamentos/informações dos veículos é também uma forma de aumentar a medida F das entidades extraídas pelo programa ExtrAuto.

Quanto aos anúncios de venda de casas e ao conjunto de testes com a ferramenta MinorThird, e de modo a aumentar os valores da medida F obtidos: a utilização de grupos nas etiquetas XML nas entidades que no CD estão colocadas sequencialmente e separadas por virgulas; o *tuning* dos algoritmos existentes na ferramenta através da alteração dos vários parâmetros e opções existentes para cada algoritmo utilizado; a criação de um CD com informação em maior quantidade e qualidade para as entidades que apresentam piores resultados globais.

Bibliografia

Aitken, J. S. (2002). Learning information extraction rules: An inductive logic programming approach. [citado na pág. 5]

Au, K. and Cheung, K. (2004). Information extraction for on-line job advertisements. [citado na pág. 16]

Baum, L. E. and Eagon, J. A. (1967). An inequality with applications to statistical estimation for probalistic functions of Markov processes and to a model for ecology. *Bull. Am. Math. Soc.*, 73:360–363. [citado na pág. 10]

Baum, L. E., Petrie, T., Soules, G., and Weiss, N. (1970). A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. *The Annals of Mathematical Statistics*, 41(1):164–171. [citado na pág. 10, 14]

Baum, L. E. and Petrie, T. P. (1966). Statistical inference for probabilistic functions of finite state Markov chains. *The Annals of Mathematical Statistics*, 37:1554–1563. [citado na pág. 10]

Baumgartner, R., Flesca, S., and Gottlob, G. (2001). Visual web information extraction with lixto. [citado na pág. 9]

Benouareth, A., Ennaji, A., and Sellami, M. (2006). Hmms with explicit state duration applied to handwritten arabic word recognition. In *ICPR '06: Proceedings of the 18th International Conference on Pattern Recognition*, pages 897–900, Washington, DC, USA. IEEE Computer Society. [citado na pág. 18]

Bikel, D. M., Miller, S., Schwartz, R., and Weischedel, R. (1997). Nymble: a high-performance learning name-finder. In *In Proceedings of the Fifth Conference on Applied Natural Language Processing*, pages 194–201. [citado na pág. 16]

Cai, J. and Liu, Z.-Q. (1999). Integration of structural and statistical information for unconstrained handwritten numeral recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 21(3):263–270. [citado na pág. 18]

Chen, M. Y., Kundu, A., and Srihari, S. N. (1993). Variable duration hidden markov model and morphological segmentation for handwritten word recognition. *Computer Vision and Pattern Recognition, 1993. Proceedings CVPR '93., 1993 IEEE Computer Society Conference on*, pages 600–601. [citado na pág. 18]

Ciravegna, F. (2001). Lp2 an adaptive algorithm for information extraction from web-related texts. In *In Proceedings of the IJCAI-2001 Workshop on Adaptive Text Extraction and Mining.* [citado na pág. 10]

Cohen, W. (2004a). *MinorThird: Methods for Identifying Names and Ontological Relations in Text using Heuristics for Inducing Regularities from Data.* http://minorthird.sourceforge.net. [citado na pág. 30, 42, 102]

Cohen, W. W. (2004b). Exploiting dictionaries in named entity extraction: Combining semi-markov extraction processes and data integration methods. In *Semi-Markov Extraction Processes and Data Integration Methods, Proceedings of KDD 2004*, pages 89–98. [citado na pág. 43]

Collins, M. (2002). Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. pages 1–8. [citado na pág. 43]

Connan, J. and Omlin, C. W. (2000). Bibliography extraction with hidden markov models. Technical report. [citado na pág. 16]

Cortes, C. and Vapnik, V. (1995). Support-vector networks. In *Machine Learning*, pages 273–297. [citado na pág. 25]

Culotta, A., Bekkerman, R., and Mccallum, A. (2004). Extracting social networks and contact information from email and the web. In *In CEAS-1.* [citado na pág. 24]

Dalianis, H. and Velupillai, S. (2010). De-identifying swedish clinical text - refinement of a gold standard and experiments with conditional random fields. [citado na pág. 25]

Darroch, J. N. and Ratcliff, D. (1972). Generalized iterative scaling for log-linear models. *The Annals of Mathematical Statistics*, 43(5):1470–1480. [citado na pág. 20]

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38. [citado na pág. 14, 18]

Dingare, S., Nissim, M., Finkel, J., Manning, C., and Grover, C. (2005). A system for identifying named entities in biomedical text: how results from two evaluations reflect on both the system and the evaluations: Conference papers. *Comp. Funct. Genomics*, 6(1-2):77–85. [citado na pág. 21]

Emms, M. (2001). A prolog based information extraction system. In *In Proceedings of the International Applications of Prolog Conference*, pages 160–167. [citado na pág. 10]

feng Lin, Y., han Tsai, T., chi Chou, W., pin Wu, K., yi Sung, T., and lian Hsu, W. (2004). A maximum entropy approach to biomedical named entity recognition. In *in Proceedings of the 4th ACM SIGKDD Workshop on Data Mining in Bioinformatics*, pages 56–61. [citado na pág. 21]

Forney, G. D. (1973). The viterbi algorithm. *Proceedings of the IEEE*, 61(3):268–278. [citado na pág. 13, 18, 24]

Freitag, D. and Kushmerick, N. (2000). Boosted wrapper induction. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence*, pages 577–583. AAAI Press. [citado na pág. 9]

Freitag, D. and Mccallum, A. K. (1999). Information extraction with hmms and shrinkage. In *In Proceedings of the AAAI-99 Workshop on Machine Learning for Information Extraction*, pages 31–36. [citado na pág. 16]

Fresko, M., Rosenfeld, B., and Feldman, R. (2005). A hybrid approach to ner by memm and manual rules. In *CIKM 05: Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 361–362, New York, NY, USA. ACM. [citado na pág. 22]

George R. Thoma, S. M. and Misra, D. (2005). Automated metadata extraction to preserve the digital contents of biomedical collections. [citado na pág. 18]

Giles, H. H. C. L., Manavoglu, E., Zha, H., Zhang, Z., and Fox, E. A. (2003). Automatic document metadata extraction using support vector machines. In *In JCDL 03: Proceedings of the 3rd ACM/IEEE-CS Joint Conference on Digital Libraries*, pages 37–48. [citado na pág. 28]

Hobbs, J. R., Bear, J., Israel, D., and Tyson, M. (1993). Fastus: A finite-state processor for information extraction from real-world text. pages 1172–1178. [citado na pág. 9]

Hoberman, R. and Durand, D. (2006). Hmm lecture notes. [citado na pág. 12, 14]

Huang, J., Zweig, G., and Padmanabhan, M. (2001). Information extraction from voicemail. In *In Proceedings of the Conference of the Association for Computational Linguistics*, pages 290–297. [citado na pág. 9]

hui Chang, C. and Lui, S.-C. (2001). Iepad: Information extraction based on pattern discovery. pages 681–688. [citado na pág. 9]

Isozaki, H. and Kazawa, H. (2002). Efficient support vector classifiers for named entity recognition. In *Proceedings of the 19th international conference on Computational linguistics*, pages 1–7, Morristown, NJ, USA. Association for Computational Linguistics. [citado na pág. 28]

Jansche, M. (2002). Named entity extraction with conditional markov models and classifiers. [citado na pág. 21]

Jaynes, E. T. (1957). Information theory and statistical mechanics. *Physical Review Online Archive (Prola)*, 106(4):620–630. [citado na pág. 19]

Jayram, T. S., Krishnamurthy, R., Raghavan, S., Vaithyanathan, S., and Zhu, H. (2006). Avatar information extraction system. *IEEE Data Engineering Bulletin*, 29:2006. [citado na pág. 10]

Kazama, J., Makino, T., Ohta, Y., and Tsujii, J. (2002). Tuning support vector machines for biomedical named entity recognition. In *In Proceedings of the ACL-02 Workshop on Natural Language Processing in the Biomedical Domain*, pages 1–8. [citado na pág. 28]

Kim, S., Yoon, J., Park, K.-M., and Rim, H.-C. (2005). Two-phase biomedical named entity recognition using a hybrid method. In *Natural Language Processing IJCNLP 2005*, volume 3651 of *Lecture Notes in Computer Science*, pages 646–657. Springer Berlin. [citado na pág. 21]

Klinger, R., Friedrich, C. M., Fluck, J., and Hofmann-apitius, M. (2007). Named entity recognition with combinations of conditional random fields. [citado na pág. 24]

Krishnamurthy, R., Li, Y., Raghavan, S., Reiss, F., Vaithyanathan, S., and Zhu, H. (2008). Systemt: a system for declarative information extraction. *SIGMOD Rec.*, 37(4):7–13. [citado na pág. 10]

Krishnan, V. and Ganapathy, V. (2005). Named entity recognition. [citado na pág. 22]

Lafferty, J. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. pages 282–289. Morgan Kaufmann. [citado na pág. 22, 43]

Leek, T. R. (1997). Information extraction using hidden markov models. [citado na pág. 16]

Li, R., Liu, L. Y., Fu, H. F., and Zheng, J. H. (2009). Application study of hidden markov model and maximum entropy in text information extraction. In *AICI '09: Proceedings of the International Conference on Artificial Intelligence and Computational Intelligence*, pages 399–407, Berlin, Heidelberg. Springer-Verlag. [citado na pág. 22]

Li, Y., Bontcheva, K., and Cunningham, H. (2005). Svm based learning system for information extraction. In *In Proceedings of Sheffield Machine Learning Workshop, Lecture Notes in Computer Science*. Springer Verlag. [citado na pág. 28]

Lucarelli, G., Vasilakos, X., and Androutsopoulos, I. (2007). Named entity recognition in greek texts with an ensemble of svms and active learning. [citado na pág. 28]

Makhoul, J., Kubala, F., Schwartz, R., and Weischedel, R. (1999). Performance measures for information extraction. In *In Proceedings of DARPA Broadcast News Workshop*, pages 249–252. [citado na pág. 29]

Malouf, R. (2002). A comparison of algorithms for maximum entropy parameter estimation. In *COLING-02: proceedings of the 6th conference on Natural language learning*, pages 1–7, Morristown, NJ, USA. Association for Computational Linguistics. [citado na pág. 24]

Mansouri, A., Affendy, L. S., and Mamat, A. (2008). A new fuzzy support vector machine method for named entity recognition. In *ICCSIT 08: Proceedings of the 2008 International Conference on Computer Science and Information Technology*, pages 24–28, Washington, DC, USA. IEEE Computer Society. [citado na pág. 28]

McCallum, A. (2003). Efficiently inducing features of conditional random fields. [citado na pág. 24]

Mccallum, A. and Freitag, D. (2000). Maximum entropy markov models for information extraction and segmentation. pages 591–598. Morgan Kaufmann. [citado na pág. 19, 21, 43]

Mcdonald, R. and Pereira, F. (2005). Identifying gene and protein mentions in text using conditional random fields. [citado na pág. 24]

Minkov, E. and Wang, R. C. (2005). Extracting personal names from emails: Applying named entity recognition to informal text. In *In HLT-EMNLP*. [citado na pág. 24]

Moens, M.-F. (2006). *Information Extraction: Algorithms and Prospects in a Retrieval Context (The Information Retrieval Series)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA. [citado na pág. 1, 25]

Packer, T., Lutes, J., Stewart, A., Embley, D., Ringger, E., and Seppi, K. (2010a). Extracting person names from diverse and noisy ocr text. Department of Computer Science Brigham Young University Provo, Utah, USA. [citado na pág. 22]

Packer, T., Lutes, J., Stewart, A., Embley, D., Ringger, E., and Seppi, K. (2010b). Extracting person names from diverse and noisy ocr text. [citado na pág. 24]

Peng, F. and McCallum, A. (2006). Information extraction from research papers using conditional random fields. *Inf. Process. Manage.*, 42(4):963–979. [citado na pág. 24]

Pietra, S. D., Pietra, V. D., and Lafferty, J. (1997). Inducing features of random fields. *IEEE Transactions on pattern analysis and machine intelligence*, 19(4):380–393. [citado na pág. 20, 24]

Pinto, D., Mccallum, A., Wei, X., and Croft, W. B. (2003). Table extraction using conditional random fields. [citado na pág. 24]

Ponomareva, N., Rosso, P., Pla, F., and Molina, A. (2006). Conditional random fields vs. hidden markov models in a biomedical named entity recognition task. [citado na pág. 24]

Rabiner, L. and Juang, B. (2003). An introduction to hidden markov models. *ASSP Magazine, IEEE*, 3(1):4–16. [citado na pág. 11, 12, 13, 18, 24]

Rabiner, L. R. (1990). A tutorial on hidden markov models and selected applications in speech recognition. pages 267–296. [citado na pág. 12, 18, 24]

Riloff, E. (1996). Automatically generating extraction patterns from untagged text. In *AAAI/IAAI, Vol. 2*, pages 1044–1049. [citado na pág. 9]

Safabakhsh, R. and Adibi, P. (2005). Nastaaligh handwritten word recognition using a continuous density variable-duration hmm. *The Arabian J. Science and Eng*, 30:95–118. [citado na pág. 18]

Saha, S. K., Narayan, S., Sarkar, S., and Mitra, P. (2010). A composite kernel for named entity recognition. *Pattern Recogn. Lett.*, 31(12):1591–1597. [citado na pág. 28]

Sarawagi, S. (2008). Information extraction. *FnT Databases*, 1(3). [citado na pág. 6, 8]

Sarawagi, S. and Cohen, W. W. (2004). Semi-markov conditional random fields for information extraction. In *In Advances in Neural Information Processing Systems 17*, pages 1185–1192. [citado na pág. 43]

Settles, B. (2004). Biomedical named entity recognition using conditional random fields and rich feature sets. In *JNLPBA '04: Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications*, pages 104–107, Morristown, NJ, USA. Association for Computational Linguistics. [citado na pág. 24]

Seymore, K., Mccallum, A., and Rosenfeld, R. (1999). Learning hidden markov model structure for information extraction. In *AAAI 99 Workshop on Machine Learning for Information Extraction*. [citado na pág. 16]

Sha, F. and Pereira, F. (2003). Shallow parsing with conditional random fields. pages 213–220. [citado na pág. 24, 43]

Soderland, S. (1999). Learning information extraction rules for semi-structured and free text. *Machine Learning*, 34(1-3):233–272. [citado na pág. 9]

Soderland, S., Fisher, D., Aseltine, J., and Lehnert, W. (1995). Crystal: Inducing a conceptual dictionary. [citado na pág. 9]

Sun, A., Naing, M.-M., Lim, E.-P., and Lam, W. (2003). Using support vector machines for terrorism information extraction. In *ISI 03: Proceedings of the 1st NSF/NIJ conference on Intelligence and security informatics*, pages 1–12, Berlin, Heidelberg. Springer-Verlag. [citado na pág. 28]

Takeuchi, K. and Collier, N. (2003). Bio-medical entity extraction using support vector machines. [citado na pág. 28]

Tang, J., Zhang, J., Yao, L., Li, J., Zhang, L., and Su, Z. (2008). Arnetminer: Extraction and mining of academic social networks. [citado na pág. 24]

Viterbi, A. (2003). Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *Information Theory, IEEE Transactions on*, 13(2):260–269. [citado na pág. 13, 18, 24]

Wong, W., Martinez, D., and Cavedon, L. (2009). Extraction of named entities from tables in gene mutation literature. [citado na pág. 28]

WorldWideWebSize (2009). `http://www.worldwidewebsize.com/`. [citado na pág. 1]

Wu, J.-h. and Zhou, J. (2008). An approach of chunk parsing and entity relation extracting to chinese based on conditional random fields model. In *ISDA '08: Proceedings of the 2008 Eighth International Conference on Intelligent Systems Design and Applications*, pages 489–494, Washington, DC, USA. IEEE Computer Society. [citado na pág. 24]

Yakushiji, A., Tateisi, Y., Miyao, Y., and ichi Tsujii, J. (2001). Event extraction from biomedical papers using a full parser. In *Pac. Symp. Biocomput*, pages 408–419. [citado na pág. 9]

Yang, Z., Lin, H., and Li, Y. (2010). Bioppisvmextractor: A protein-protein interaction extractor for biomedical literature using svm and rich feature sets. *J. of Biomedical Informatics*, 43(1):88–96. [citado na pág. 28]

Yu, S. (2009). Hidden semi-markov models. *Artificial Intelligence*, 174(2):215–243. [citado na pág. 17, 18]

Yun, B.-H. (2010). Hmm-based korean named entity recognition for information extraction. In Zhang, Z. and Siekmann, J., editors, *Knowledge Science, Engineering and Management*, volume 4798 of *Lecture Notes in Computer Science*, pages 526–531. Springer Berlin / Heidelberg. [citado na pág. 16]

Zhang, N. R. (2001). Hidden markov models for information extraction. [citado na pág. 16]